



**TUGAS AKHIR - TF 145565**

**PERANCANGAN PROTOTIPE *ANTENNA TRACKER*  
BERBASIS *GLOBAL POSITIONING SYSTEM (GPS)***

**PURWANTO PAMUJI  
NRP. 2411 031 058**

**Dosen Pembimbing  
Andi Rahmadiansah, ST., MT.**

**PROGRAM STUDI DIII METROLOGI DAN INSTRUMENTASI  
JURUSAN TEKNIK FISIKA  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2015**



**FINAL PROJECT - TF 145565**

**PROTOTYPE DESIGN OF ANTENNA TRACKER BASED  
ON GLOBAL POSITIONING SYSTEM (GPS)**

**PURWANTO PAMUJI**  
**NRP. 2411 031 058**

**Supervisor**  
**Andi Rahmadiansah, ST., MT.**

**INSTRUMENTATION AND METROLOGY DIII STUDY PROGRAM**  
**DEPARTMENT OF ENGINEERING PHYSICS**  
**Faculty of Industrial Technology**  
**Institut Teknologi Sepuluh Nopember**  
**Surabaya 2015**

**LEMBAR PENGESAHAN**  
**PERANCANGAN PROTOTYPE ANTENNA TRACKER**  
**BERBASIS GLOBAL POSITIONING SYSTEM (GPS)**

**TUGAS AKHIR**

**Oleh:**

**PURWANTO PAMUJI**

**NRP 2411 031 058**

**Surabaya, Januari 2015**

**Mengetahui/Menyetujui**

**Pembimbing**



**Andi Rahmadiansah, ST., MT.**

**NIP : 19790517 200312 1 002**



**Ketua Jurusan**  
**Teknik Fisika FTI-ITS**

**Dr. Ir. Totok Soehartanto, DEA**

**NIP : 19650309 199002 1 001**





**LEMBAR PENGESAHAN**  
**PERANCANGAN PROTOTIPE ANTENNA TRACKER**  
**BERBASIS GLOBAL POSITIONING SYSTEM (GPS)**

**TUGAS AKHIR**

Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Ahli Madya  
pada  
Bidang Studi Rekayasa Instrumentasi  
Program Studi DIII Metrologi dan Instrumentasi  
Jurusan Teknik Fisika  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember Surabaya

Oleh:

**PURWANTO PAMUJI**  
**NRP 2411 031 058**

Disetujui oleh Tim Penguji Tugas Akhir:

1. Andi Rahmadiansah, ST., MT. .... (Pembimbing)
2. Ir. Yerri Susatio, MT. .... (Penguji I)
3. Ir. Tutug Dhanardono, MT. .... (Penguji II)
4. Ir. Heri Joestiono, MT. .... (Penguji III)
5. Detak Yan Pratama, ST., MSc. .... (Penguji IV)

**SURABAYA**  
**Januari 2015**



## PERANCANGAN PROTOTIPE ANTENNA TRACKER BERBASIS GLOBAL POSITIONING SYSTEM (GPS)

Nama Mahasiswa : PURWANTO PAMUJI  
NRP : 2411 031 058  
Jurusan : Teknik Fisika  
Dosen Pembimbing : Andi Rahmadiansah, ST., MT.

### Abstrak

*Global positioning system* adalah perangkat yang semakin banyak diadopsi dengan tujuan agar dapat memantau posisi dari objek yang diinginkan. Beberapa implementasi dari GPS adalah penunjuk rute yang akan dilalui, pemantauan jarak jauh kendaraan serta penjejak satelit. Selain dapat diimplementasikan dalam bidang keamanan, GPS juga dapat digunakan untuk mengoptimalkan jangkauan serta kualitas komunikasi antara *payload* dengan *ground station*. Standar GPS NMEA dengan format keluaran yang dapat menyampaikan informasi letak suatu objek, dapat dimanfaatkan untuk mengoptimalkan komunikasi yaitu dengan merancang *antenna tracker*. *Antenna tracker* dapat melacak letak objek sesuai dengan koordinat yang diterima. Dengan mengetahui koordinat tetap dari *ground station*, sistem dapat berjalan dengan memperoleh variabel baru dari GPS *transmitter*. Koordinat dari GPS *transmitter* dikalkulasi dengan *haversine formula* untuk memperoleh nilai *azimuth* dan sudut elevasi antara *transmitter* dan *ground station*. Penelitian ini menghasilkan presentase *error* pembacaan ketinggian dan jarak dari GPS masing-masing 5,7% dan 17,3%. Sedangkan rata-rata *error* dari aktuator posisi *pan* dan *tilt* masing-masing  $1,8^0$  dan  $1,3^0$ .

Kata Kunci : *Global Positioning System, Haversine Formula, Azimuth, Elevation*







## **PROTOTYPE DESIGN OF ANTENNA TRACKER BASED ON GLOBAL POSITIONING SYSTEM (GPS)**

*Student Name* : PURWANTO PAMUJI  
*NRP* : 2411 031 058  
*Department* : Engineering Physics  
*Supervisor* : Andi Rahmadiansah, ST., MT.

### **Abstract**

*Global positioning system is a device that is increasingly being adopted in order to monitor the position of the desired object. Some implementations of GPS is pointing the route to be traversed, remote monitoring of vehicles and satellite tracking. Besides being able to be implemented into the field of security, GPS can also be used to optimize the range and quality of communication between the payloads and the ground station. The NMEA GPS standard with output format that can convey the location information of an object, can be used to optimize the communication by designing the antenna tracker. Antenna tracker can track the location of the object in accordance with the coordinates received. By knowing the fixed coordinates of the ground station, the system can be operated with the new variable gain from a GPS transmitter. The coordinates of the GPS transmitter is calculated with haversine formula to obtain the value of the azimuth and elevation angle between the transmitter and the ground station. This research resulted in the percentage error reading from the GPS altitude and distance respectively 5.7% and 17.3%. While the average error of the pan and tilt actuator position respectively  $1.8^{\circ}$  and  $1.3^{\circ}$ .*

*Keywords* : Global Positioning System, Haversine Formula, Azimuth, Elevation



## DAFTAR ISI

HALAMAN JUDUL .....	i
LEMBAR PENGESAHAN .....	v
ABSTRAK .....	ix
ABSTRACT .....	xi
KATA PENGANTAR .....	xiii
DAFTAR ISI .....	xv
DAFTAR GAMBAR .....	xvii
DAFTAR TABEL .....	xix
 BAB I. PENDAHULUAN .....	 1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan .....	2
1.5 Sistematika Laporan .....	2
 BAB II. TINJAUAN PUSTAKA .....	 5
2.1 <i>Global Positioning System</i> .....	5
2.2 <i>Azimuth</i> dan Elevasi .....	7
2.3 Mikrokontroler .....	9
2.4 <i>Servo Motor</i> .....	9
2.5 <i>Haversine Formula</i> .....	10
 BAB III. METODOLOGI PENELITIAN .....	 13
3.1 Perancangan Sistem .....	13
3.1.1 Perancangan Stasiun Pemancar atau <i>Uplink</i> .....	15
3.1.2 Perancangan Stasiun Penerima atau <i>Downlink</i> .....	16
3.1.3 Perancangan Aktuator <i>Antenna Tracker</i> .....	17
3.2 Pengujian Sistem .....	19
3.2.1 Pengujian Sensor <i>Global Positioning</i> <i>System</i> .....	19
3.2.2 Pengujian <i>Servo Motor</i> .....	20

BAB IV. PENGUJIAN DAN ANALISA .....	23
4.1    Pengujian .....	23
4.1.1    Pengujian Sensor <i>Global Positioning System</i> .....	23
4.1.2    Pengujian <i>Servo Motor</i> .....	27
4.2    Analisa Data .....	34
4.2.1    Analisa Data <i>Global Positioning System</i> (GPS) .....	34
4.2.2    Analisa Data Motor <i>Servo</i> .....	35
 BAB V. PENUTUP .....	 37
5.1    Kesimpulan .....	37
5.2    Saran .....	37
 DAFTAR PUSTAKA .....	 39



## DAFTAR TABEL

Tabel	4.1	Tabel Koordinat Pengujian Sensor <i>Global Positioning System</i> .....	23
Tabel	4.2	Tabel Koordinat Pembacaan Sensor <i>Global Positioning System</i> .....	24
Tabel	4.3	Tabel Hasil Perbandingan Sensor <i>Global Positioning System</i> .....	27
Tabel	4.4	Pengujian Motor <i>Servo</i> Sudut <i>Pan</i> .....	27
Tabel	4.5	Pengujian Motor <i>Servo</i> Sudut <i>Tilt</i> .....	28
Tabel	4.6	Data Sampling Pengujian Terbang dengan <i>Antenna Tracker</i> .....	30
Tabel	4.7	Data Sampling Pengujian Terbang Tanpa Menggunakan <i>Antenna Tracker</i> .....	32



## DAFTAR GAMBAR

Gambar 2.1	<i>Global Positioning System</i> .....	5
Gambar 2.2	Sudut <i>azimuth</i> dan elevasi .....	8
Gambar 2.3	Sudut <i>Pan</i> dan Sudut <i>Tilt</i> .....	8
Gambar 2.4	Contoh gambar Mikrokontroler .....	9
Gambar 2.5	<i>Servo Motor</i> .....	10
Gambar 2.6	<i>Distance Point to Point</i> .....	10
Gambar 2.7	<i>Great Circle Distance</i> .....	11
Gambar 3.1	Diagram alir penelitian .....	14
Gambar 3.2	Diagram alir stasiun pemancar .....	15
Gambar 3.3	Diagram alir stasiun penerima .....	16
Gambar 3.4	Diagram alir aktuator <i>antenna tracker</i> .	18
Gambar 4.1	Pemetaan Titik Uji dan Hasil Pembacaan Sensor .....	24
Gambar 4.2	Visualisasi 3 Dimensi Titik Pengujian Sensor <i>Global Positioning System</i> .....	25
Gambar 4.3	Visualisasi 3 Dimensi Titik Pembacaan Sensor <i>Global Positioning System</i> .....	25
Gambar 4.4	Visualisasi 3 Dimensi Titik Pengujian dan Pembacaan Sensor <i>Global Positioning System</i> .....	26
Gambar 4.5	Visualisasi 2 Dimensi Data Sampling menggunakan <i>Antenna Tracker</i> .....	31
Gambar 4.6	Visualisasi 3 Dimensi Data Sampling menggunakan <i>Antenna Tracker</i> .....	31
Gambar 4.7	Visualisasi 2 Dimensi Data Sampling tanpa menggunakan <i>Antenna Tracker</i> ....	33
Gambar 4.8	Visualisasi 3 Dimensi Data Sampling tanpa menggunakan <i>Antenna Tracker</i> ....	33





## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

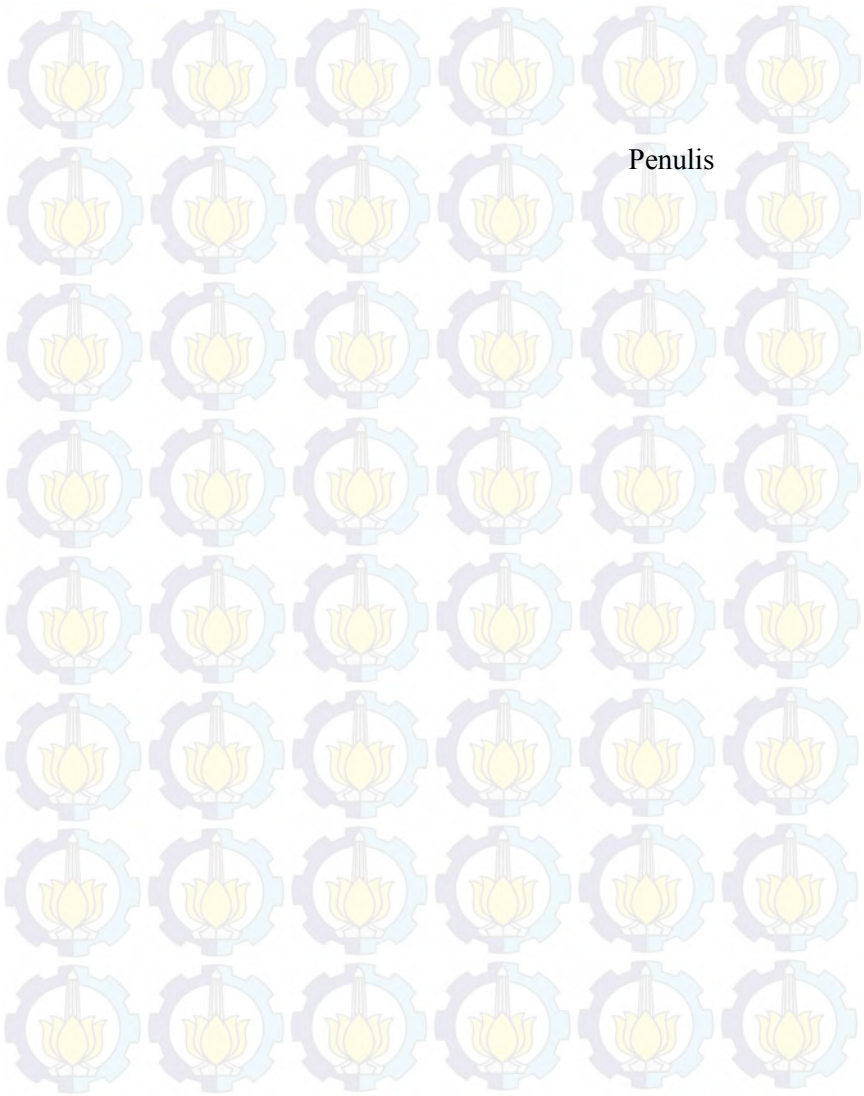
Puji syukur kehadiran Allah SWT yang senantiasa melimpahkan rahmat dan hidayah-Nya serta shalawat dan salam kepada Nabi Muhammad SAW, hingga terselesaikannya Tugas Akhir beserta Laporan Tugas Akhir yang berjudul PERANCANGAN PROTOTYPE ANTENNA TRACKER BERBASIS GLOBAL POSITIONING SYSTEM (GPS).

Penulis telah banyak memperoleh bantuan dari berbagai pihak dalam penyelesaian Tugas Akhir dan Laporan Tugas Akhir ini. Penulis mengucapkan terima kasih kepada :

1. Allah SWT yang telah melimpahkan rahmad dan hidayah-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir dan Laporan Tugas Akhir ini.
2. Nabi Muhammad SAW yang telah menjadi panutan bagi seluruh alam sehingga penulis dapat mengerti bagaimana hidup yang baik dan benar di dunia dan di hadapan Allah.
3. Kedua orang tua, ibu dan ayah yang telah membesarkan, mendidik, menuntun dan mengarahkan pada kebaikan.
4. Saudara dan saudari, Bambang Siswoyo, Dewi Handayani, Erna Gustiawati, dan Hariyanto Wibowo.
5. Bapak Andi Rahmadiansah selaku pembimbing dan Kepala Laboratorium Simulasi dan Komputasi yang telah ikhlas dan penuh kesabaran dalam menghadapi, mengarahkan dan membimbing saya.
6. Keluarga besar Labkom-E205, teman-teman angkatan 2011 Teknik Fisika dan rekan-rekan yang telah memberi dukungan moril dan materil.

Penulis menyadari bahwa penulisan pada tugas akhir ini belum sempurna. Namun penulis berharap semoga tulisan Tugas Akhir yang telah diselesaikan memiliki manfaat yang besar untuk diri sendiri terutama, untuk Jurusan Teknik Fisika FTI-ITS, dan InshaAllah untuk bangsa dan negara tercinta. Dan semoga generasi penerus Tugas Akhir dengan materi yang sama dapat menjadikan

Tugas Akhir ini salah satu referensi yang dapat dipertanggungjawabkan.



## DAFTAR PUSTAKA

- [1] Si, H., & Aung, Z. M. (2011). Position Data Acquisition from NMEA Protocol of Global Positioning System. *International Journal of Computer and Electrical Engineering*, 353-357.
- [2] Karol, R. (2012). *High Altitude Ballooning for Space and Atmospheric Observation*. Institute for Space Studies.
- [3] Hubner, P. (...). *Ground Station Tracking and Telemetry System*. ....
- [4] Nichat, M. K., & Chopde, N. R. (2013). Landmark Based Shortest Path Detection by Using A\* and Haversine Formula. *International Journal of Innovative Research in Computer and Communication Engineering*, 298-302.
- [5] Bullock, R. (2007). *Great Circle Distances and Bearings Between Two Locations*. ....
- [6] Xu, G. (2007). *GPS Theory, Algorithms and Applications 2nd*. Postdam: ...
- [7] Heselton, R. R. (1998). *Elevation Effects on GPS Positional Accuracy*. Blacksburg: Virginia Polytechnic Institute and State University.
- [8] Barnes, M., & Pangborn, H. (2012). *Design and Launch of a High Altitude Balloon*. .... Penn State Department of Mechanical Engineering.
- [9] Megasakti, M. C. (2010). *Rancang Bangun Auto Tracking dengan menggunakan Microcontroller, GPS, SAT Finder dan Digital Compass untuk Sinkronisasi Azimuth Antena Terhadap Satelit Cakrawarta-2*. Depok: Universitas Indonesia.

- [10] Michael, K., McNamee, A., & Michael, M. (2006). The Emerging Ethics of Humancentric GPS Tracking and Monitoring. *International Conference on Mobile Business* (p. . . .). Copenhagen: IEEE Computer Society.
- [11] Carlson, D. G., & Clay, D. D. (. . .). The Earth Model—Calculating Field Size and Distances between Points using GPS Coordinates. . . ., 1-4.



## BIODATA PENULIS



Penulis bernama Purwanto Pamuji, lahir pada tanggal 9 April 1993 di Ruteng-Flores, NTT. Riwayat pendidikan formal penulis dimulai pada tahun 1999 di SDK St.Agnes Ruteng IV selanjutnya pada tahun 2005 penulis melanjutkan pendidikan di SMP NEGERI 1 Ruteng. Pada tahun 2008, penulis melanjutkan pendidikan di SMA NEGERI 1 Ruteng kemudian dilanjutkan di SMA Muhammadiyah 7 Surabaya pada tahun 2009. Setelah lulus dari pendidikan menengah atas, penulis melanjutkan pendidikan di Institut Teknologi Sepuluh Nopember Surabaya dengan prodi D3 Metrologi dan Instrumentasi, Jurusan Teknik Fisika dengan nomor mahasiswa 2411031058. Menjadi salah satu admin dari laboratorium di Teknik Fisika Laboratorium Simulasi dan Komputasi sebagai penanggung jawab bidang programming. Penulis menjalani penelitian akhir yaitu “Perancangan Prototipe Antenna Tracker Berbasis Global Positioning System (GPS)” dibimbing oleh Bapak Andi Rahmadiansah, ST., MT.

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Sekarang ini penyampaian informasi melalui televisi atau stasiun pemantau lain yang menggunakan media transmisi satelit telah menjadi salah satu kebutuhan hidup masyarakat. Kebutuhan masyarakat akan informasi dan hiburan meningkat seiring perkembangan teknologi. Tetapi permasalahan yang muncul saat ini adalah ketika antena pemancar memiliki posisi yang berubah-ubah terhadap antena penerima atau objek yang sedang diamati seperti *radiosonde* serta *rocket payload*.

Selain kondisi geografis, cuaca serta spesifikasi komponen yang digunakan, letak pemancar yang terus berpindah memiliki dampak yang sangat besar terhadap proses transmisi data. Atas dasar setiap antena memiliki pancaran sinyal yang berbeda serta penyebaran sinyal tersebut statis, maka *user* diharuskan untuk mengembangkan inovasi baru. Oleh karena itu, untuk mengatasi hal tersebut maka perlu dirancang suatu sistem peralatan yang memungkinkan antena penerima dapat menerima seluruh data yang dikirim melalui antena pemancar.

Pembuatan sistem penjejak pada antena penerima adalah salah satu solusi untuk permasalahan tersebut, dimana antena dapat bergerak pada sudut *azimuth* dan elevasi sehingga terjadi sinkronisasi optimal antara antena pemancar dan antena penerima. Komponen yang digunakan pada penelitian ini adalah GPS yang berfungsi untuk menentukan posisi antena pemancar, motor DC, modem dan mikrokontroler.

Pada penelitian sebelumnya, antena penjejak digunakan untuk sistem keamanan dimana antena penjejak melacak letak kendaraan atau benda. Sedangkan pada penelitian ini, antena penjejak difungsikan agar transmisi pada *radiosonde*, *rocket payload*, serta stasiun cuaca menjadi optimal.

## 1.2 Rumusan Masalah

Adapun beberapa permasalahan yang terdapat dalam pengerjaan tugas akhir ini adalah sebagai berikut:

1. Bagaimana merancang *hardware* atau kondisi objek untuk mencapai tujuan dari penelitian ini?
2. Bagaimana merancang serta membangun sistem antena penjejak otomatis?
3. Bagaimana mengintegrasikan bagian-bagian yang ada agar sistem dapat berjalan maksimal dan optimal?

## 1.3 Batasan Masalah

Adapun beberapa batasan masalah pada penelitian ini adalah:

1. Spesifikasi komponen yang digunakan terbatas pada komponen standar laboratorium.
2. Pengujian dilakukan pada skala laboratorium dengan tinggi maksimal 100 meter serta radius 100 meter.
3. Sistem penjejak menggunakan GPS sebagai nilai acuan, yaitu *latitude*, *longitude* dan *altitude*.

## 1.4 Tujuan

Adapun tujuan dilakukannya penelitian ini adalah:

1. Merancang sistem serta pemilihan komponen yang tepat sebagai bagian dari perancangan sistem antena penjejak.
2. Merancang dan membangun sistem agar antena penerima dapat memperoleh nilai sudut *azimuth* dan sudut elevasi dari pemancar.
3. Mengintegrasikan komponen-komponen yang telah ada sebagai acuan dari perancangan sistem antena penjejak.

## 1.5 Sistematika laporan

Secara sistematis, penyusunan laporan tugas akhir ini tersusun dalam lima bab dengan penjelasan sebagai berikut:



## **BAB I Pendahuluan**

Bab ini berisi latar belakang, perumusan masalah, batasan masalah, tujuan penelitian, dan sistematika laporan.

## **BAB II Tinjauan Pustaka**

Bab ini berisi mengenai teori-teori penunjang yang terkait dalam penulisan tugas akhir.

## **BAB III Metodologi Penelitian**

Bab ini akan dijelaskan mengenai langkah-langkah yang telah dilakukan dalam penelitian.

## **BAB IV Pengujian dan Analisa**

Bab ini akan ditampilkan data dan analisa hasil pengujian beserta pembahasannya.

## **BAB V Penutup**

Bab ini berisi tentang kesimpulan pokok dari seluruh rangkaian penelitian yang telah dilakukan dan saran yang dapat dijadikan sebagai pengembangan penelitian selanjutnya.





## BAB II

### TINJAUAN PUSTAKA

#### 2.1 *Global Positioning System*

GPS adalah *global positioning system* (sistem posisi global) menggunakan teknologi satelit. Prinsip kerja GPS yaitu mengukur jarak antara *receiver* (penerima) dan beberapa satelit pendukung. Posisi dari beberapa satelit tersebut diperkirakan dan dikirim oleh sinyal GPS kepada pengguna. Melalui beberapa posisi (dari satelit) dan jarak dari *receiver* (penerima) terhadap satelit, posisi dari *receiver* (penerima) dapat diketahui. Perubahan posisi juga dapat diketahui, berdasarkan pada kecepatan dari *receiver*. Hal terpenting dari aplikasi GPS adalah posisi dan navigasi.



**Gambar 2.1** *Global Positioning System*

Keluaran sensor *global positioning system* memiliki standar atau enkripsi tersendiri sebagai identitas lokasi terbaca. Standar format yang digunakan pada GPS adalah format NMEA (*National Marine Electronics Association*). NMEA telah mengembangkan perangkat yang dapat menghubungkan berbagai perlengkapan elektronik pada dunia kelautan.

Standar NMEA digunakan atas dasar kelengkapan informasi yang diberikan. Data keluaran NMEA pada GPS berupa posisi, kecepatan dan waktu GMT. Beberapa format NMEA adalah GPGLL, GPGGA, GPGSA, GPRMC, GPVTG, dan GPZDA. Header “GP” pada setiap format NMEA berarti *global positioning*.

Standar format keluaran yang digunakan pada penelitian ini adalah format GPGGA dan GPRMC. Penggunaan ke dua format tersebut berdasar kebutuhan sistem yang ada. Selain variabel yang dibutuhkan, penggunaan dua variabel tersebut mengoptimalkan waktu pemrosesan sistem yang digunakan.

*\$GPGGA,123519,4807.038,N,01131.000,E,1,08,0.9,545.4,M,46.9,M,,\*47* (1.1)

Dari persamaan 1.1 diketahui bahwa:

<i>GGA</i>	<i>Global Positioning System Fix Data</i>
<i>123519</i>	<i>Fix taken at 12:35:19 UTC</i>
<i>4807.038,N</i>	<i>Latitude 48 deg 07.038' N</i>
<i>01131.000,E</i>	<i>Longitude 11 deg 31.000' E</i>
<i>1</i>	<i>Fix quality: 0 = invalid</i>
	<i>1 = GPS fix (SPS)</i>
	<i>2 = DGPS fix</i>
	<i>3 = PPS fix</i>
	<i>4 = Real Time Kinematic</i>
	<i>5 = Float RTK</i>
	<i>6 = estimated (dead reckoning)</i>
	<i>7 = Manual input mode</i>
	<i>8 = Simulation mode</i>
<i>08</i>	<i>Number of satellites being tracked</i>
<i>0.9</i>	<i>Horizontal dilution of position</i>
<i>545.4,M</i>	<i>Altitude, Meters, above mean sea level</i>
<i>46.9,M</i>	<i>Height of geoid (mean sea level) above WGS84 Ellipsoid</i>
<i>(empty field)</i>	<i>Time in seconds since last DGPS update</i>
<i>(empty field)</i>	<i>DGPS station ID number</i>
<i>*47</i>	<i>The checksum data, always begins with *</i>

\$GPRMC,123519,A,4807.038,N,01131.000,E,022.4,084.4,23039  
4,003.1,W\*6A (1.2)

Dari persamaan 1.2 diketahui bahwa:

<i>RMC</i>	<i>Recommended Minimum sentence C</i>
<i>123519</i>	<i>Fix taken at 12:35:19 UTC</i>
<i>A</i>	<i>Status A=active or V=Void.</i>
<i>4807.038,N</i>	<i>Latitude 48 deg 07.038' N</i>
<i>01131.000,E</i>	<i>Longitude 11 deg 31.000' E</i>
<i>022.4</i>	<i>Speed over the ground in knots</i>
<i>084.4</i>	<i>Track angle in degrees True</i>
<i>230394</i>	<i>Date - 23rd of March 1994</i>
<i>003.1,W</i>	<i>Magnetic Variation</i>
<i>*6A</i>	<i>The checksum data, always begins with *</i>

## 2.2 Azimuth dan Elevasi

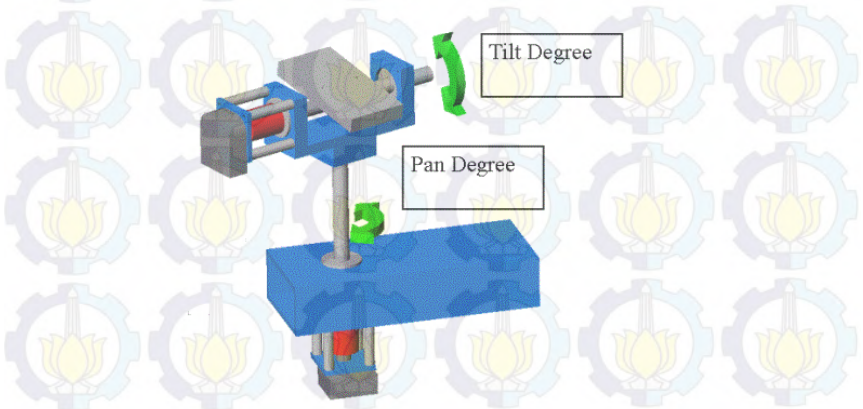
Antena memiliki peran penting dalam bidang komunikasi. Salah satu parameter penting dari antena adalah diameter antena. Semakin besar diameter antena, semakin besar pula *gain* yang diperoleh. Disamping itu, diperoleh juga *beamwidth* yang sempit. Untuk memaksimalkan nilai yang menjadi parameter pada antena tersebut, dua hal yang harus diperhatikan adalah sudut *azimuth* dan sudut elevasi. Sudut *azimuth* adalah sudut putar arah horizontal, dimana arah utara sebagai referensi sudut nol. Sedangkan sudut elevasi adalah sudut yang dibentuk oleh bidang horizontal dengan arah vertikal antena.





**Gambar 2.2** Sudut *azimuth* dan elevasi

Nilai dari sudut *azimuth* dan sudut elevasi yang diperoleh, ditransmisikan menuju aktuator agar stasiun penerima dapat menjejak posisi dari pemancar. Dimana sudut *azimuth* diimplementasikan pada aktuator *pan* dan sudut elevasi diimplementasikan pada aktuator *tilt*.



**Gambar 2.3** Sudut *Pan* dan Sudut *Tilt*

### 2.3 Mikrokontroler

Mikrokontroler adalah suatu *integrated circuit* (IC) yang pada umumnya digunakan untuk mengontrol alat tertentu misalnya sistem kontrol mobil atau sistem permesinan serta lain sebagainya. Berbeda dengan mikroprosesor yang pada umumnya digunakan pada PC (*Personal Computer*) yang hanya digunakan untuk memproses suatu data input yang diberikan sesuai dengan namanya yaitu mikroprosesor. Mikrokontroler dirancang untuk digunakan pada alat, sistem elektronik maupun mekanik, dan telah mempunyai memori, *clock*, *I/O port*, dan lain sebagainya dalam satu sistem yaitu MinSis (*Minimum System*) berbeda dengan mikroprosesor yang membutuhkan tambahan komponen lain agar suatu sistem komputer tersebut dapat beroperasi sesuai dengan yang kita inginkan.



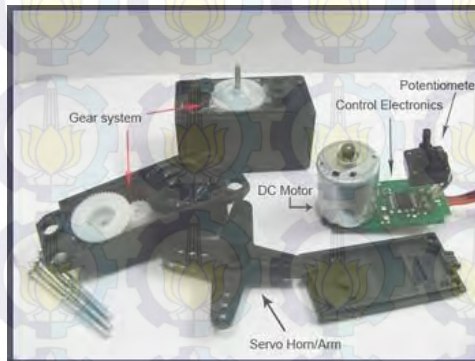
**Gambar 2.4** Contoh gambar Mikrokontroler

Pada penelitian ini suatu kontroler dibutuhkan guna mengendalikan pulsa pada motor servo agar membentuk sudut *azimuth* dan elevasi yang bertujuan untuk mengarahkan antena penerima menuju antena pemancar.

### 2.4 Servo Motor

Servo motor adalah salah satu motor DC (dalam beberapa kasus, servo merupakan motor AC) dengan komponen-komponen yang menjadikannya sebagai motor DC. Motor servo terdiri dari motor DC, potensiometer, roda gigi, dan sirkuit elektrik.

Kombinasi antara rangkaian elektrik dan potensiometer akan memudahkan pengguna mengendalikan servo. Pada penelitian ini, motor servo digunakan sebagai aktuator. Untuk mencapai tujuan penelitian ini, digunakan dua buah motor servo. Motor servo pertama digunakan sebagai aktuator pada sudut *azimuth* atau *bearing*, dan motor servo kedua digunakan sebagai aktuator pada sudut elevasi.



**Gambar 2.5** *Servo Motor*

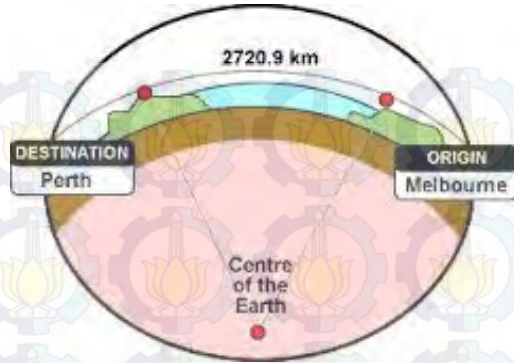
## 2.5 Haversine Formula

*Haversine formula* adalah persamaan yang penting dalam sistem navigasi. Persamaan ini digunakan untuk memperoleh jarak melingkar (*great circle distance*) antara dua titik dalam lingkup garis lintang dan garis bujur.



**Gambar 2.6** *Distance Point to Point*





**Gambar 2.7** Great Circle Distance

Secara umum, *haversine formula* dapat ditulis :

$$c = \sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\psi_2 - \psi_1}{2}\right)} \quad (2.1)$$

$$D = 2r \sin^{-1}(c) \quad (2.2)$$

Berikut adalah persamaan yang digunakan untuk menghitung *bearing* atau sudut *azimuth* antara titik pertama dan titik kedua :

$$\beta_1 = \tan^{-1}\left(\frac{\cos\phi_2\sin(\psi_2 - \psi_1)}{\cos\phi_1\sin\phi_2 - \sin\phi_1\cos\phi_2\cos(\psi_2 - \psi_1)}\right) \quad (2.3)$$

$$\beta = 180 / \pi * \beta_1 \quad (2.4)$$

Pada persamaan 2.1, persamaan 2.2, persamaan 2.3, dan persamaan 2.4 diketahui bahwa :

D = jarak melingkar antara dua titik

$\beta$  = sudut *azimuth* antara titik pertama dan titik kedua

r = jari-jari bumi (6371 km atau 3958.756 mil)

$\phi_1, \phi_2$  = satuan radian posisi lintang atau *latitude* titik pertama dan titik kedua

$\psi_1, \psi_2$  = satuan radian posisi bujur atau *longitude* titik pertama dan titik kedua





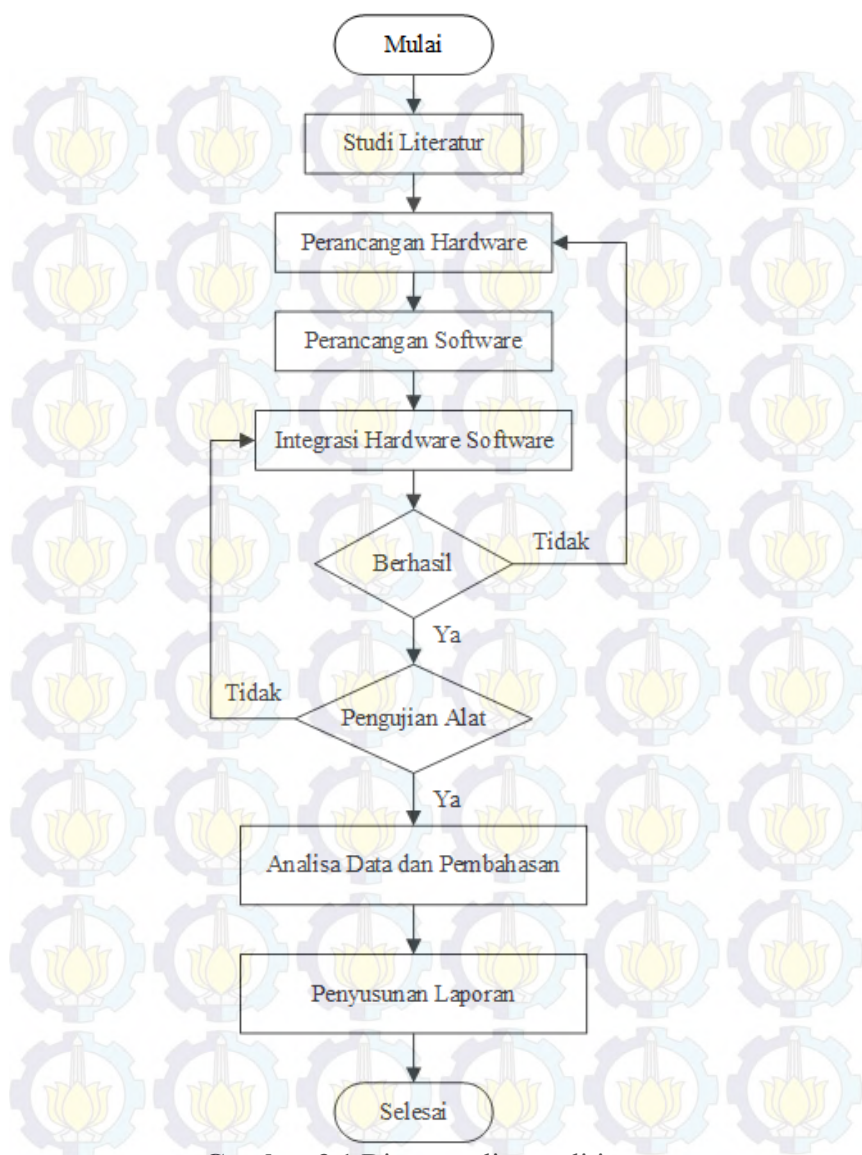
## BAB III METODOLOGI PENELITIAN

### 3.1 Perancangan Sistem

Penelitian ini dilakukan sebagai salah satu cara untuk mengoptimalkan komunikasi antara satelit pemancar atau objek dengan stasiun penerima. Penelitian ini berupa perancangan protipe *antenna tracker* dengan *global positioning system*. Parameter *antenna tracker* ini berupa posisi lintang serta bujur dari objek yang diamati. Dan keluaran yang dihasilkan adalah sudut *azimuth* dan sudut elevasi.

Pada penelitian ini, perubahan nilai lintang dan nilai bujur berpengaruh terhadap posisi *pan* atau sudut *azimuth* serta posisi *tilt* atau sudut elevasi. Perubahan koordinat akan berdampak pada jarak pemancar dan penerima, dimana jarak merupakan salah satu variabel yang digunakan untuk mendapatkan sudut *azimuth* dan sudut elevasi.

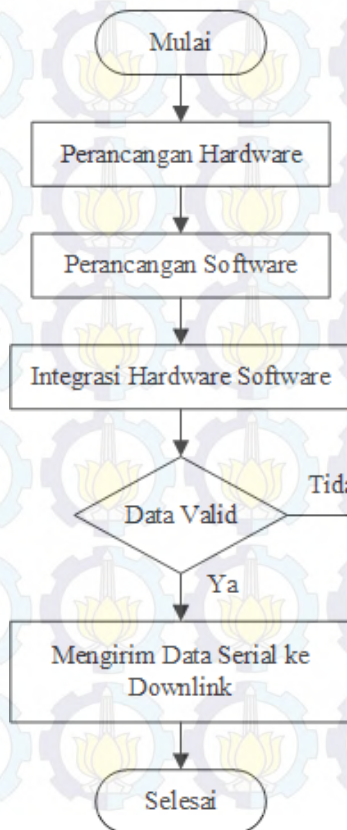
Untuk mendapatkan hasil yang optimal, terlebih dahulu dilakukan studi literatur mengenai GPS serta algoritma yang dapat diimplementasikan sesuai dengan karakteristik GPS. Setelah itu, mencari persamaan model matematis sebagai acuan pada perancangann prototipe *antenna tracker*. Kemudian merancang sistem kontrol pada *antenna tracker* berdasarkan pada persamaan matematis 2.4. Uji kinerja sistem diperlukan untuk mengetahui apakah sistem yang dirancang telah berhasil agar komunikasi dapat berjalan lebih optimal. Untuk alur selengkapnya dapat dilihat pada diagram-alir pada gambar 3.1:



**Gambar 3.1** Diagram alir penelitian

### 3.1.1 Perancangan Stasiun Pemancar atau *Uplink*

Perancangan stasiun pemancar bertujuan untuk mendapatkan koordinat letak dari objek pemancar yang telah diintegrasikan dengan sensor *global positioning system*. Keluaran yang dari sistem ini berupa koordinat lintang (*latitude*), koordinat bujur (*bujur*) dan letak ketinggian di atas permukaan laut dari stasiun pemancar (*altitude*). Untuk alur perancangan sistem stasiun pemancar atau *uplink* dapat dilihat pada diagram-alir pada gambar 3.2:

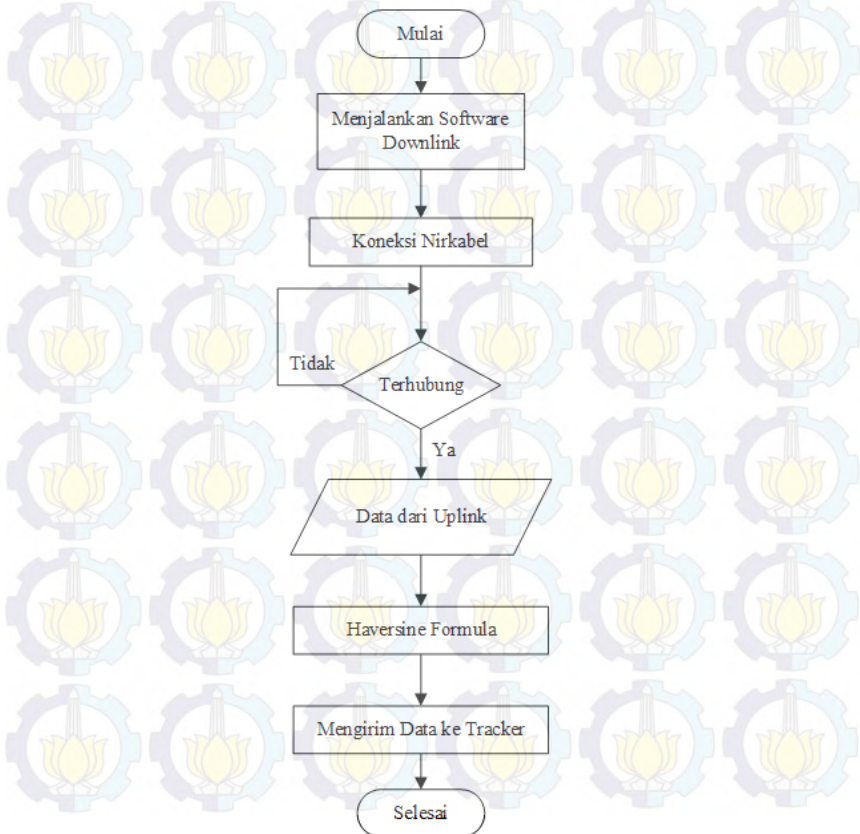


**Gambar 3.2** Diagram alir stasiun pemancar



### 3.1.2 Perancangan Stasiun Penerima atau *Downlink*

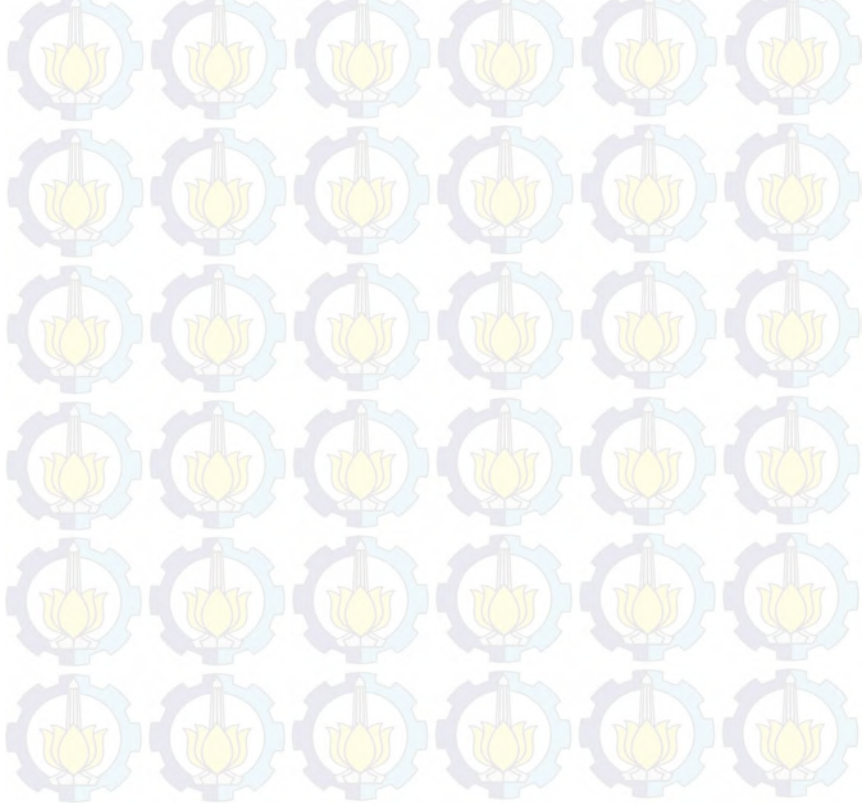
Perancangan stasiun penerima bertujuan untuk memproses data yang diterima dari stasiun pemancar, berupa koordinat letak stasiun pemancar dengan *haversine formula*. Dengan *haversine formula* (persamaan 2.4), koordinat letak stasiun pemancar dan koordinat letak stasiun penerima akan diproses untuk mendapatkan sudut *azimuth* dan sudut elevasi antara kedua objek. Untuk alur perancangan sistem stasiun penerima atau *downlink* dapat dilihat pada diagram alir pada gambar 3.3:

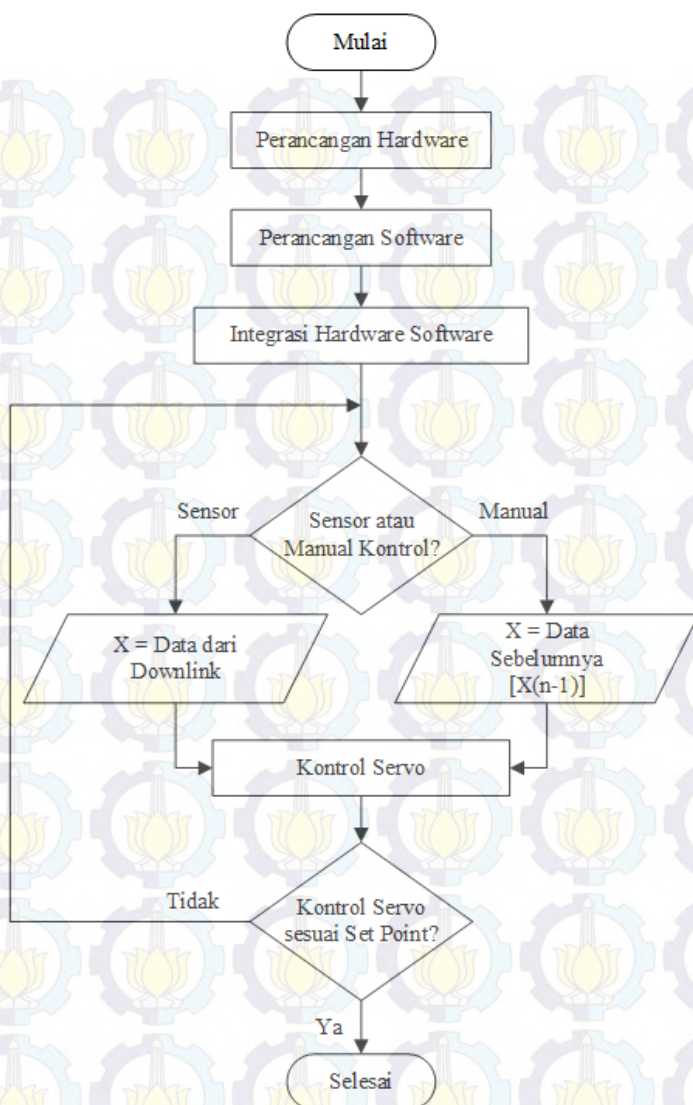


**Gambar 3.3** Diagram alir stasiun penerima

### 3.1.3 Perancangan Aktuator *Antenna Tracker*

Perancangan aktuator ini bertujuan untuk mengeksekusi atau mengimplementasikan hasil perhitungan yang telah diperoleh menggunakan *haversine formula* (persamaan 2.4) ke dalam aktuator. Nilai yang menjadi dasar untuk menjalankan aktuator ini adalah nilai dari hasil perhitungan yaitu sudut *azimuth* dan nilai sudut elevasi. Kedua nilai tersebut akan dikonversi menjadi pulsa menggunakan mikrokontroler agar dapat menggerakkan motor servo. Untuk alur perancangan aktuator *antenna tracker* dapat dilihat pada diagram alir pada gambar 3.4:





Gambar 3.4 Diagram alir aktuator antenna tracker

### 3.2 Pengujian Sistem

Pengujian sistem ini bertujuan untuk menguji perangkat keras yang telah diintegrasikan dengan perangkat lunak. Pengujian dilakukan untuk mendapatkan hasil dari keluaran sensor yang digunakan, hasil perhitungan matematis serta keluaran pada aktuator. Berikut adalah persamaan yang digunakan untuk memperoleh *error* pembacaan dari sensor yang digunakan:

$$\varepsilon_1 = \frac{\sum SK}{n} \quad (3.1)$$

$$\varepsilon_2 = \frac{\sum SJ}{n} \quad (3.2)$$

$$\varepsilon\% = \frac{(D_2 - D_1)}{D_1} \times 100 \quad (3.3)$$

Dari persamaan 3.1, persamaan 3.2 dan persamaan 3.3 diketahui:

$\varepsilon\%$  = presentase *error*.

$\varepsilon_1$  = nilai rata-rata *error* pembacaan ketinggian (mdpl) dari sensor *global positioning system*.

$\varepsilon_2$  = nilai rata-rata *error* pembacaan garis lintang dan garis bujur dari sensor *global positioning system*.

$D_1$  = data asli.

$D_2$  = data hasil.

SK = selisih ketinggian.

SJ = selisih jarak.

n = jumlah data.

#### 3.2.1 Pengujian Sensor *Global Positioning System*

Pengujian sensor *global positioning system* dilakukan dengan beberapa tahapan berikut ini:

1. Menempatkan *uplink* pada tempat terbuka. Hal ini bertujuan untuk mengurangi interferensi frekuensi lain seperti bangunan, medan magnet dari elektronik, maupun frekuensi alat komunikasi lainnya agar sensor mendapatkan sinyal yang dipancarkan oleh satelit.



2. Memastikan koneksi antara *uplink* dan *downlink*.
3. Menempatkan sensor pada beberapa titik untuk menguji tingkat akurasi dari sensor.
4. Membandingkan hasil keluaran sensor dengan pengukuran secara langsung. Variabel yang dapat digunakan pada pengujian ini adalah jarak antara titik pertama penempatan sensor dengan titik berikutnya. Nilai toleransi yang ditetapkan untuk hasil pembacaan sensor ini adalah kurang dari 20%.

### 3.2.2 Pengujian Servo Motor

Motor servo merupakan aktuator yang digunakan untuk mengeksekusi hasil perhitungan matematis yang telah diperoleh. Motor servo berfungsi untuk mengarahkan antena menuju posisi *uplink* agar komunikasi antara *uplink* dan *downlink* menjadi lebih optimal. Berikut adalah persamaan yang digunakan untuk memperoleh *error* dari aktuator:

$$\varepsilon_3 = \frac{\sum ST_p}{n} \quad (3.4)$$

$$\varepsilon_4 = \frac{\sum ST_t}{n} \quad (3.5)$$

Dari persamaan 3.4 dan persamaan 3.5 diketahui bahwa:

$\varepsilon_3$  = *error* motor servo posisi *pan*.

$\varepsilon_4$  = *error* motor servo posisi *tilt*.

$ST_t$  = sudut *real* yang terbaca menggunakan busur.

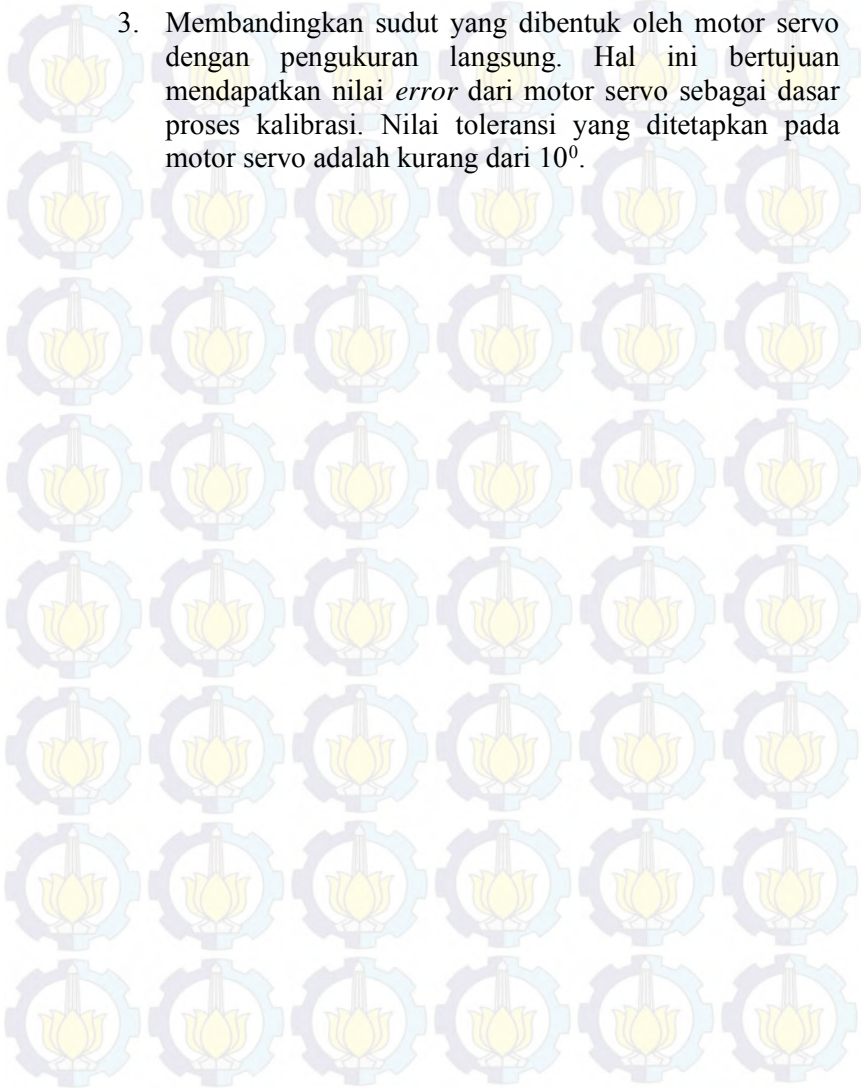
$ST_p$  = sudut *real* yang terbaca menggunakan busur.

$n$  = jumlah data.

Pengujian motor servo dilakukan dengan beberapa tahapan berikut ini:

1. Memastikan *wiring* antara motor servo dan mikrokontroler serta memastikan koneksi antara aktuator dengan *downlink*.

2. Mengirimkan nilai *integer* berupa sudut ( $0^0$ - $360^0$ ) menuju mikrokontroler.
3. Membandingkan sudut yang dibentuk oleh motor servo dengan pengukuran langsung. Hal ini bertujuan mendapatkan nilai *error* dari motor servo sebagai dasar proses kalibrasi. Nilai toleransi yang ditetapkan pada motor servo adalah kurang dari  $10^0$ .





## BAB IV PENGUJIAN DAN ANALISA

### 4.1 Pengujian

Prototipe perangkat keras *antenna tracker* diintegrasikan dengan perangkat lunak yang telah ada, termasuk didalamnya perhitungan matematis *haversine formula* (persamaan 2.4). Pengujian dilakukan dalam dua tahap, yaitu:

#### 4.1.1 Pengujian Sensor *Global Positioning System*

Pengujian sensor *global positioning system* dilakukan untuk mengetahui tingkat akurasi dan besar *error* dari sensor yang digunakan. Pengujian dilakukan di beberapa koordinat berbeda yang telah ditentukan sebelumnya. Hal ini bertujuan untuk membandingkan hasil keluaran sensor dengan koordinat yang telah ditentukan.

**Tabel 4.1** Tabel Koordinat Pengujian Sensor *Global Positioning System*

No.	Tanggal	Lintang	Bujur	Ketinggian (mdpl)	Jarak (m) [X(n-1)]
1.	14-12-2014	-7.618256	112.846936	17	0
2.	14-12-2014	-7.618256	112.847120	17	20
3.	14-12-2014	-7.618440	112.847120	17	20
4.	14-12-2014	-7.618440	112.846936	17	20

Tabel 4.1 merupakan tabel yang koordinat lintang, bujur serta ketinggiannya telah ditentukan. Koordinat-koordinat tersebut ditentukan menggunakan meteran serta *software google earth* sebagai acuan untuk mendapat nilai garis lintang dan bujur. Koordinat tersebut kemudian digunakan untuk menguji tingkat akurasi sensor yang digunakan sebelum *uplink* diuji dengan diterbangkan menggunakan balon atmosfer.



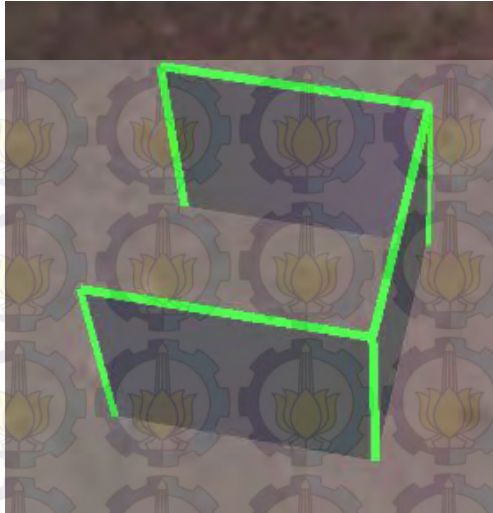
**Tabel 4.2** Tabel Koordinat Pembacaan Sensor *Global Positioning System*

No.	Tanggal	Lintang	Bujur	Ketinggian (mdpl)	Jarak (m) [X(n-1)]
1.	14-12-2014	-7.618256	112.846936	18,3	0
2.	14-12-2014	-7.618217	112.847133	18,7	22,1
3.	14-12-2014	-7.618407	112.847114	16,8	21,7
4.	14-12-2014	-7.618426	112.846916	18,1	22,3

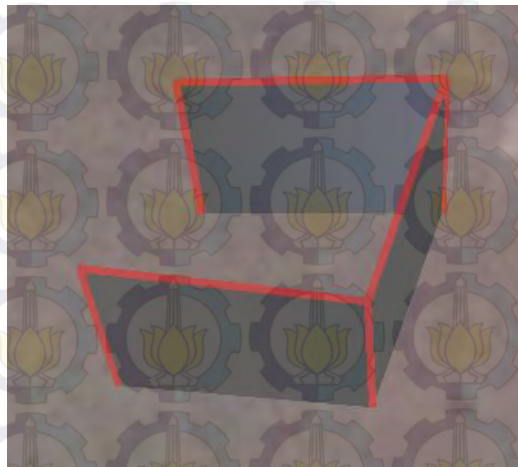
Tabel 4.2 merupakan hasil pembacaan sensor. Pengujian dilakukan dengan menempatkan sensor di koordinat yang telah ditentukan pada tabel 4.1. Perbandingan antara koordinat yang menjadi dasar uji dengan data hasil pembacaan sensor terdapat pada tabel 4.3. Pemetaan koordinat pengujian data dapat dilihat pada gambar 4.1.



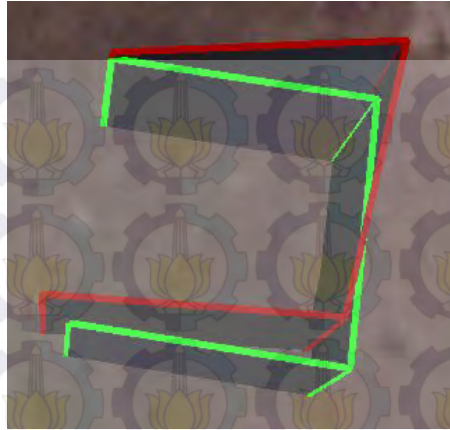
**Gambar 4.1** Pemetaan Titik Uji dan Hasil Pembacaan Sensor



**Gambar 4.2** Visualisasi 3 Dimensi Titik Pengujian Sensor *Global Positioning System*



**Gambar 4.3** Visualisasi 3 Dimensi Titik Pembacaan Sensor *Global Positioning System*



**Gambar 4.4** Visualisasi 3 Dimensi Titik Pengujian dan Pembacaan Sensor *Global Positioning System*

Pada gambar 4.1, pemetaan koordinat pengujian divisualisasikan. Terdapat delapan titik koordinat pada pengujian. Empat titik koordinat digambarkan berwarna hijau atau koordinat yang diinisialisasikan dengan huruf kapital merupakan koordinat yang telah ditentukan sebagai koordinat pengujian. Empat titik koordinat digambarkan berwarna merah atau koordinat yang diinisialisasikan dengan huruf non-kapital merupakan koordinat hasil pembacaan sensor *global positioning system*.

Hasil perbandingan dikalkulasi menggunakan persamaan 3.1, persamaan 3.2 dan persamaan 3.3 sehingga diperoleh nilai *error* dari sensor GPS. Selain itu, diperoleh pula rata-rata *error* ketinggian dan jarak, masing masing 1,1 meter dan 1,5 meter. Sedangkan presentase *error* dari ketinggian dan jarak masing-masing 5,7% dan 17,3%. Hasil perhitungan terdapat pada tabel 4.3.

**Tabel 4.3** Tabel Hasil Perbandingan Sensor *Global Positioning System*

No.	Selisih Ketinggian (mdpl)	Selisih Jarak (meter)	Error Ketinggian (%)	Error Jarak (%)
1.	1,3	0	7,6	0
2.	1,7	2,1	10	10,5
3.	0,2	1,7	1,1	8,5
4.	1,1	2,3	6,4	11,5

#### 4.1.2 Pengujian Servo Motor

Pengujian ini dilakukan dengan dua tahap yaitu pengujian motor servo pada posisi *pan* (*yaw* atau sumbu *z*) dan pengujian motor servo pada posisi *tilt* (*pitch* atau sumbu *y*). Posisi *pan* sendiri merupakan implementasi dari sudut *azimuth*, sedangkan posisi *tilt* merupakan implementasi dari sudut elevasi.

Sudut yang dibentuk oleh kedua motor servo mempunyai rentang yang berbeda. Motor servo pada posisi *pan* membentuk sudut  $0^0$  hingga sudut  $360^0$ . Motor servo pada sudut *tilt* membentuk sudut  $0^0$  sampai sudut  $180^0$ . Posisi *pan* dan *tilt* ditunjukkan pada gambar 2.3.

**Tabel 4.4** Pengujian Motor Servo Sudut *Pan*

No.	Set Point (derajat)	PWM (microsecond)	Sudut Terbaca (derajat)	Error (%)
1.	0	600	0	0
2.	30	620	32	6,6
3.	60	640	61	1,6
4.	90	660	92	2,2
5.	120	678	117	2,5
6.	180	718	178	1,2
7.	240	752	242	0,8
8.	270	770	272	0,7
9.	330	805	327	0,9
10.	360	823	359	0,2



Tabel 4.4 merupakan tabel yang berisi data pengujian motor servo yang digunakan pada posisi *pan*. Pengujian dilakukan dengan menentukan *set point* berjumlah sepuluh data berbeda. Pada tahap ini, proses pembacaan dilakukan menggunakan busur derajat. Validasi ini bertujuan untuk memperoleh besar *error* dari motor servo yang digunakan. Besar *error* diperoleh dengan menggunakan persamaan 3.4. Sedangkan rata-rata *error* yang dihasilkan oleh motor servo posisi *pan* adalah  $1,8^0$ .

**Tabel 4.5** Pengujian Motor Servo Sudut *Tilt*

No.	<i>Set Point</i> (derajat)	PWM ( <i>microsecond</i> )	Sudut Terbaca (derajat)	<i>Error</i> (%)
1.	0	620	0	0
2.	30	936	28	6,6
3.	45	1095	47	4,4
4.	60	1253	58	3,3
5.	90	1570	91	1,1
6.	120	1886	121	0,8
7.	135	2045	134	0,7
8.	150	2203	148	1,3
9.	165	2361	167	1,2
10.	180	2520	180	0

Tabel 4.5 merupakan tabel yang berisi data pengujian motor servo yang digunakan pada posisi *tilt*. Pengujian dilakukan dengan menentukan *set point* berjumlah sepuluh data berbeda. Pada tahap ini, proses pembacaan dilakukan menggunakan busur derajat. Validasi ini bertujuan untuk memperoleh besar *error* dari motor servo yang digunakan. Besar *error* diperoleh dengan menggunakan persamaan 3.5. Sedangkan rata-rata *error* yang dihasilkan oleh motor servo posisi *tilt* adalah  $1,3^0$ .

Setelah pengujian 4.1.1 dan pengujian 4.1.2 dilakukan, diperoleh tingkat akurasi dari sensor serta *error* dari aktuator. Kedua nilai tersebut akan digunakan untuk validasi atau sebagai nilai koreksi pada data selanjutnya.

Setelah proses validasi dan kalibrasi dilakukan, pengujian terakhir pada penelitian ini adalah uji terbang. Pengujian ini dilakukan dengan menerbangkan *uplink* atau *payload* menggunakan balon atmosfer. Hal ini bertujuan untuk melihat respon sistem yang telah ada. Selain menguji sistem *antenna tracker* yang telah ada, pengujian juga dilakukan dengan transmisi tanpa menggunakan sistem *antenna tracker*. Hal ini bertujuan untuk memperoleh tujuan dari dilakukannya penelitian ini.

**Tabel 4.6** Data Sampling Pengujian Terbang dengan *Antenna Tracker*

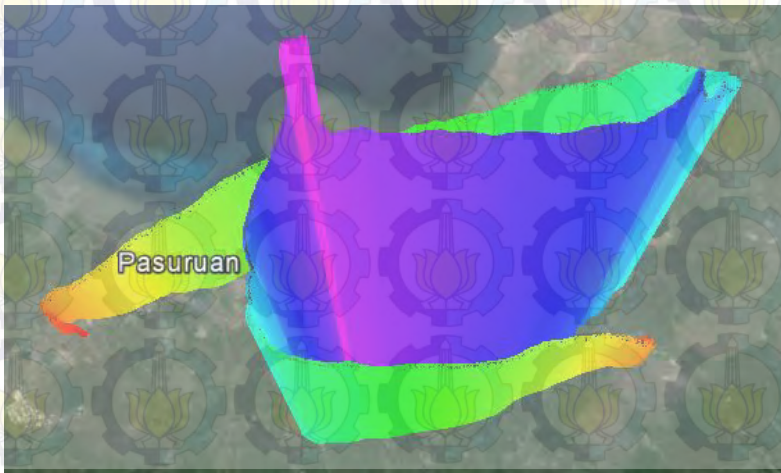
No.	Waktu	Lintang	Bujur	Ketinggian (mdpl)
1.	12:47:09	-7.618259	112.846947	4,88
2.	12:48:19	-7.616929	112.846359	510,84
3.	12:49:29	-7.614873	112.845377	890,63
4.	12:50:38	-7.613354	112.843536	1322,22
5.	12:51:48	-7.612209	112.842407	1766,01
6.	12:52:58	-7.610083	112.842834	2229,31
7.	12:54:07	-7.609563	112.843327	2689,25
8.	12:55:17	-7.610161	112.846072	3172,97
9.	12:56:26	-7.611663	112.847867	3647,54
10.	12:57:36	-7.614098	112.851086	4139,49
11.	12:58:46	-7.618016	112.855456	4636,31
12.	12:59:55	-7.620564	112.859276	5131,61
13.	13:01:04	-7.622923	112.862694	5631,18
14.	13:02:14	-7.625433	112.869237	6137,45
15.	13:03:23	-7.627396	112.874661	6656,83
16.	13:04:32	-7.630099	112.879427	7189,32
17.	13:05:42	-7.633033	112.882846	7736,74
18.	13:06:51	-7.635729	112.886551	8309,15
19.	13:08:00	-7.640188	112.894116	8870,29
20.	13:09:09	-7.642893	112.902176	9440,88
21.	13:10:18	-7.647008	112.910006	10002,32
22.	13:11:27	-7.652079	112.918674	10561,02
23.	13:12:36	-7.659709	112.926432	11131.6
24.	13:13:45	-7.668023	112.934257	11626.6
25.	13:14:54	-7.676481	112.942092	12178.28
26.	13:16:03	-7.684798	112.952609	12699.8
27.	13:17:12	-7.691679	112.959716	13214.3
28.	13:18:21	-7.696703	112.968406	13721.18
29.	13:19:30	-7.703706	112.975352	14180.52
30.	13:20:39	-7.709994	112.980684	14650.21
31.	13:21:47	-7.712918	112.985399	15165.63
32.	13:22:56	-7.717434	112.987947	15695.07



Data pada tabel 4.6 merupakan data sampling dari total keseluruhan 5255 data. Data pada tabel 4.6 divisualisasikan dalam bentuk 2 dimensi (tampak atas) dan 3 dimensi menggunakan *software google earth*. Hasil visualisasi data terdapat pada gambar 4.5 dan gambar 4.6.



**Gambar 4.5** Visualisasi 2 Dimensi Data Sampling menggunakan *Antenna Tracker*



**Gambar 4.6** Visualisasi 3 Dimensi Data Sampling menggunakan *Antenna Tracker*

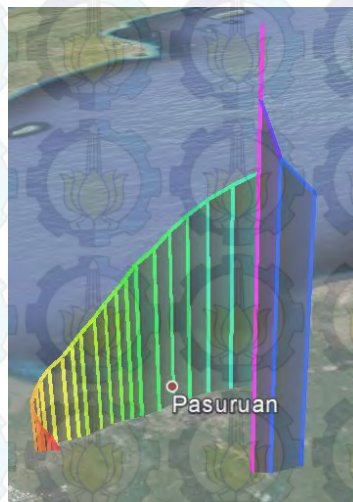


**Tabel 4.7** Data Sampling Pengujian Terbang Tanpa Menggunakan *Antenna Tracker*

No.	Waktu	Lintang	Bujur	Ketinggian (mdpl)
1.	12:47:09	-7.618259	112.846947	4.87
2.	12:48:19	-7.616929	112.846359	510.84
3.	12:49:29	-7.614873	112.845377	890.62
4.	12:50:38	-7.613354	112.843536	1322.22
5.	12:51:48	-7.612209	112.842407	1766.01
6.	12:52:58	-7.610083	112.842834	2229.30
7.	12:54:07	-7.609563	112.843327	2689.25
8.	12:55:17	-7.610161	112.846072	3172.96
9.	12:56:26	-7.611663	112.847867	3647.54
10.	12:57:36	-7.614098	112.851086	4139.48
11.	12:58:46	-7.618016	112.855456	4636.31
12.	12:59:55	-7.620564	112.859276	5131.61
13.	13:01:04	-7.622923	112.862694	5631.18
14.	13:02:14	-7.625433	112.869237	6137.45
15.	13:03:23	-7.627396	112.874661	6656.83
16.	13:04:32	-7.630099	112.879427	7189.31
17.	13:05:42	-7.633033	112.882846	7736.73
18.	13:06:51	-7.635729	112.886551	8309.15
19.	13:08:00	-7.640188	112.894116	8870.28
20.	13:09:09	-7.642893	112.902176	9440.87
21.	13:10:18	-7.647008	112.910006	10002.31
22.	13:11:27	-7.652079	112.918674	10561.01
23.	13:12:36	-7.659709	112.926432	11131.60
24.	13:13:45	-7.668023	112.934257	11626.59
25.	14:02:44	-7.694601	112.877777	19728.78
26.	14:04:37	-7.696108	112.874031	17076.11
27.	14:06:35	-7.702578	112.877446	14785.84
28.	14:08:13	-7.711239	112.887822	13150.90



**Gambar 4.7** Visualisasi 2 Dimensi Data Sampling tanpa menggunakan *Antenna Tracker*



**Gambar 4.8** Visualisasi 3 Dimensi Data Sampling tanpa menggunakan *Antenna Tracker*

Data pada tabel 4.7 merupakan data sampling dari total keseluruhan 1622 data. Data pada tabel 4.7 divisualisasikan dalam bentuk 2 dan 3 dimensi menggunakan *software google earth*. Hasil visualisasi data terdapat pada gambar 4.7 dan gambar 4.8.

## 4.2 Analisa Data

Analisa data dapat dilakukan setelah pengujian perangkat keras dan perangkat lunak telah terintegrasi dengan baik.

### 4.2.1 Analisa Data *Global Positioning System* (GPS)

Pengujian sensor *global positioning system* dapat dilakukan dengan menentukan menempatkan sensor pada koordinat awal atau pada koordinat yang telah ditentukan. Hasil pembacaan sensor yang terdapat pada tabel 4.2 dibandingkan dengan data koordinat awal pada tabel 4.1 menghasilkan nilai koreksi yang terdapat pada tabel 4.3.

Pada tabel 4.3, terdapat nilai rata-rata *error* pembacaan sensor. Kesalahan pembacaan dapat disebabkan oleh beberapa hal berikut ini:

1. Perubahan *temperature* dan tekanan awan serta volume debu pada lapisan troposfer (0 km – 50 km diatas permukaan laut) dapat menjadi penyebab yang dapat mengganggu sinyal GPS.
2. Kondisi cuaca yang mendung pada saat pengujian sistem, dapat menyebabkan partikel-partikel pada lapisan ionosfer (50 km – 500 km diatas permukaan bumi) terionisasi (bermuatan). Kondisi tersebut dapat mengganggu sinyal GPS sehingga menjadi salah satu penyebab *error* dalam penentuan lokasi.
3. Kondisi lingkungan pengujian sistem yang berada disekitar bangunan, pepohonan serta interferensi peralatan elektronika, dapat menjadi penyebab kesalahan pembacaan pada modul GPS.
4. Orbit (lintasan) yang dilalui oleh satelit yang berdampak pada jumlah satelit referensi modul GPS *receiver*. Semakin banyak satelit yang dapat diakses oleh *receiver*, semakin akurat informasi lokasi yang diterima.

Dengan persamaan 3.3, nilai presentase *error* dari ketinggian dan presentase *error* dari jarak hasil pembacaan diperoleh. Presentase *error* yang dihasilkan dapat ditoleransi dikarenakan:



1. Presentase *error* ketinggian dari hasil pembacaan bernilai kurang dari nilai toleransi yang ditetapkan yaitu 20% pada ketinggian kurang dari 100 meter.
2. Presentase *error* jarak dari hasil pembacaan bernilai kurang dari nilai toleransi yang ditetapkan yaitu 20% pada jarak kurang dari 50 meter.

#### 4.2.2 Analisa Data Motor Servo

Pengujian dilakukan dengan memberikan nilai awal atau *set point* berupa nilai sudut yang akan dikonversi menjadi nilai PWM oleh mikrokontroler menuju motor servo. Data pada tabel 4.4 dan tabel 4.5 merupakan data pengujian motor servo berupa *set point* serta keluaran dari motor servo yang digunakan. Ketidakesesuaian antara *set point* yang diberikan dengan keluaran motor servo disebabkan beberapa hal berikut:

1. Roda gigi yang terdapat pada motor servo tidak dapat membentuk sudut sesuai dengan *set point* yang ditentukan.
2. Hasil konversi antara *set point* dan PWM yang tidak sesuai. Setelah nilai dikonversi, terdapat nilai yang menghasilkan tipe data *double*, dimana tipe data *double* tidak dapat digunakan sebagai nilai referensi untuk mengontrol motor servo. Pembulatan terjadi ketika tipe data *double* yang dihasilkan pada bahasa pemrograman C. Hal tersebut menjadi salah satu penyebab kesalahan eksekusi pada aktuator.

Dengan data hasil pengujian motor servo yang diperoleh, keluaran yang dihasilkan dapat ditoleransi. Hal tersebut dikarenakan:

1. Nilai rata-rata *error* motor servo pada posisi *pan* dari hasil pengujian bernilai kurang dari nilai toleransi yang ditetapkan yaitu  $10^0$ .
2. Nilai rata-rata *error* motor servo pada posisi *tilt* dari hasil pengujian bernilai kurang dari nilai toleransi yang ditetapkan yaitu  $10^0$ .





## **BAB V**

### **PENUTUP**

#### **5.1 Kesimpulan**

Dari hasil penelitian yang dilakukan telah didapatkan kesimpulan sebagai berikut:

1. Prototipe *antenna tracker* berbasis *global positioning system* (GPS) telah berhasil dilakukan dan menghasilkan aktuator yang dapat berorientasi pada sudut *azimuth* dan sudut elevasi sesuai dengan koordinat lokasi yang diberikan.
2. Prototipe *antenna tracker* berbasis *global positioning system* (GPS) berhasil mengendalikan aktuator sudut *azimuth* dan sudut elevasi dengan *error* yang dapat ditoleransi.
3. Presentase *error* dengan pengujian ketinggian kurang dari 100 meter bernilai kurang dari nilai toleransi maksimal yaitu 20%.
4. Presentase *error* dengan pengujian jarak atau radius kurang dari 50 meter bernilai kurang dari nilai toleransi maksimal yaitu 20%.
5. Nilai rata-rata *error* motor servo pada posisi *pan* dan posisi *tilt* bernilai kurang dari nilai toleransi maksimal yaitu 10°.

#### **5.2 Saran**

Penelitian ini masih memerlukan pengembangan agar memperoleh hasil yang lebih baik. Saran untuk pengembangan penelitian ini adalah sebagai berikut:

1. Pengembangan perangkat keras agar tumpuan dapat menopang antena dengan kapasitas yang besar serta beban yang merata.
2. Pengembangan *antenna tracker* menggunakan persamaan matematis yang memiliki tingkat akurasi yang lebih tinggi seperti *vincenty formula* dan *geographical distance*.
3. Pengembangan *antenna tracker* dengan *receive signal strength indicator* (RSSI) untuk mengantisipasi kondisi atmosfer bumi sehingga sistem dapat tetap berjalan dengan jangkauan yang lebih jauh antara pemancar dan penerima.



## Lampiran A

Pada lampiran berikut ini adalah data pengujian terbang menggunakan *antenna tracker* dengan balon udara.

**Tabel A.1** Data pengujian *antenna tracker*

No	Waktu	Latitude	Longitude	Altitude (m)
1	12:47:09	-7.618259	112.846947	4.88
2	12:47:11	-7.618241	112.846912	10.36
3	12:47:12	-7.618189	112.846881	15.24
4	12:47:13	-7.618124	112.846874	21.64
5	12:47:14	-7.618086	112.846876	28.35
6	12:47:15	-7.618098	112.846864	35.05
7	12:47:16	-7.618108	112.846842	41.76
8	12:47:18	-7.618059	112.846731	52.43
9	12:47:19	-7.618011	112.846722	58.83
10	12:47:20	-7.617978	112.846751	66.45
11	12:47:21	-7.617976	112.846772	74.07
12	12:47:22	-7.618001	112.846744	81.99
13	12:47:23	-7.618013	112.846681	89.61
14	12:47:25	-7.617903	112.846591	105.77
15	12:47:26	-7.617856	112.846606	114.91
16	12:47:27	-7.617863	112.846609	123.44
17	12:47:28	-7.617913	112.846589	133.5
18	12:47:29	-7.617944	112.846562	143.56
19	12:47:30	-7.617918	112.846527	152.4
20	12:47:32	-7.617801	112.846509	167.64
21	12:47:33	-7.617796	112.846509	175.26
22	12:47:34	-7.617821	112.846507	182.27
23	12:47:35	-7.617844	112.846472	188.67
24	12:47:36	-7.617834	112.846414	194.16



Lampiran - A2

<b>Tabel A.1(Lanjutan) Data pengujian <i>antenna tracker</i></b>				
<b>25</b>	12:47:37	-7.617794	112.846382	200.56
<b>26</b>	12:47:38	-7.617754	112.846391	208.48
<b>27</b>	12:47:40	-7.617753	112.846417	226.16
<b>28</b>	12:47:41	-7.617764	112.846386	234.39
<b>29</b>	12:47:42	-7.617766	112.846314	241.1
<b>30</b>	12:47:43	-7.617748	112.846256	247.8
<b>31</b>	12:47:44	-7.617706	112.846254	254.51
<b>32</b>	12:47:45	-7.617658	112.846294	261.52
<b>33</b>	12:47:47	-7.617624	112.846339	277.67
<b>34</b>	12:47:48	-7.617649	112.846299	286.21
<b>35</b>	12:47:49	-7.617649	112.846241	295.35
<b>36</b>	12:47:50	-7.617611	112.846224	303.89
<b>37</b>	12:47:51	-7.617566	112.846254	311.2
<b>38</b>	12:47:52	-7.617544	112.846282	319.13
<b>39</b>	12:47:54	-7.617569	112.846244	332.84
<b>40</b>	12:47:55	-7.617549	112.846189	339.55
<b>41</b>	12:47:56	-7.617508	112.846156	346.86
<b>42</b>	12:47:57	-7.617461	112.846169	354.18
<b>43</b>	12:47:58	-7.617451	112.846204	361.49
<b>44</b>	12:47:59	-7.617468	112.846222	370.64
<b>45</b>	12:48:01	-7.617431	112.846192	385.27
<b>46</b>	12:48:02	-7.617379	112.846204	392.89
<b>47</b>	12:48:03	-7.617354	112.846237	400.81
<b>48</b>	12:48:04	-7.617349	112.846249	409.04
<b>49</b>	12:48:05	-7.617349	112.846226	416.97
<b>50</b>	12:48:06	-7.617341	112.846214	423.98
<b>51</b>	12:48:08	-7.617293	112.846292	436.78
<b>52</b>	12:48:09	-7.617301	112.846316	444.7

<b>Tabel A.1(Lanjutan) Data pengujian <i>antenna tracker</i></b>				
<b>53</b>	12:48:10	-7.617298	112.846311	452.02
<b>54</b>	12:48:11	-7.617248	112.846287	459.64
<b>55</b>	12:48:12	-7.617169	112.846277	465.43
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
<b>2756</b>	13:40:05	-7.735744	112.960039	22891.7
<b>2757</b>	13:40:06	-7.735781	112.959987	22898.4
<b>2758</b>	13:40:07	-7.735814	112.959894	22904.2
<b>2759</b>	13:40:09	-7.735856	112.959657	22917.61
<b>2760</b>	13:40:10	-7.735894	112.959574	22924.62
<b>2761</b>	13:40:11	-7.735963	112.959514	22929.49
<b>2762</b>	13:40:12	-7.736048	112.959464	22936.2
<b>2763</b>	13:40:13	-7.736121	112.959412	22943.21
<b>2764</b>	13:40:14	-7.736171	112.959352	22949.61
<b>2765</b>	13:40:16	-7.736176	112.959211	22961.19
<b>2766</b>	13:40:17	-7.736176	112.959149	22968.2
<b>2767</b>	13:40:18	-7.736206	112.959102	22975.21
<b>2768</b>	13:40:19	-7.736263	112.959067	22981.31
<b>2769</b>	13:40:20	-7.736334	112.959037	22987.41
<b>2770</b>	13:40:21	-7.736408	112.959001	22994.72
<b>2771</b>	13:40:22	-7.736459	112.958944	23002.04
<b>2772</b>	13:40:24	-7.736469	112.958736	23014.53
<b>2773</b>	13:40:25	-7.736464	112.958604	23021.24
<b>2774</b>	13:40:26	-7.736491	112.958489	23027.94
<b>2775</b>	13:40:27	-7.736546	112.958397	23033.13
<b>2776</b>	13:40:28	-7.736616	112.958322	23039.22
<b>2778</b>	13:40:29	-7.736676	112.958242	23045.62
<b>2779</b>	13:40:32	-7.736706	112.957846	23063.61

Lampiran - A4

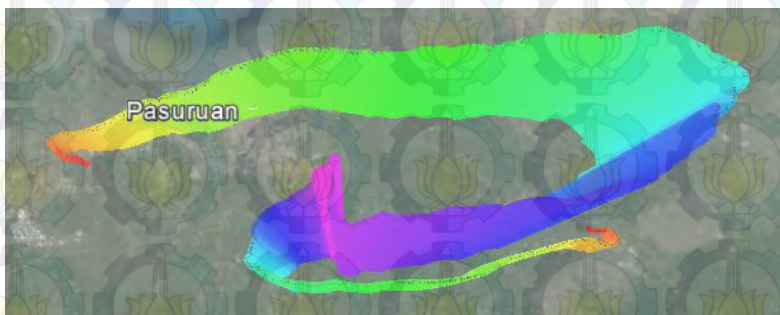
<b>Tabel A.1(Lanjutan) Data pengujian <i>antenna tracker</i></b>				
<b>2780</b>	13:40:33	-7.736694	112.957704	23070.62
<b>2781</b>	13:40:34	-7.736699	112.957586	23077.63
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
<b>5232</b>	14:27:46	-7.751248	112.969276	227.08
<b>5233</b>	14:27:47	-7.751214	112.969266	216.71
<b>5234</b>	14:27:48	-7.751186	112.969249	206.65
<b>5235</b>	14:27:49	-7.751156	112.969236	196.6
<b>5236</b>	14:27:51	-7.751078	112.969241	178
<b>5237</b>	14:27:52	-7.751049	112.969229	168.86
<b>5238</b>	14:27:53	-7.751019	112.969229	160.02
<b>5239</b>	14:27:54	-7.750986	112.969214	151.79
<b>5240</b>	14:27:55	-7.750963	112.969206	142.95
<b>5241</b>	14:27:56	-7.750923	112.969202	134.11
<b>5242</b>	14:27:58	-7.750849	112.969191	117.96
<b>5243</b>	14:27:59	-7.750803	112.969176	110.03
<b>5244</b>	14:28:00	-7.750766	112.969161	100.89
<b>5245</b>	14:28:01	-7.750724	112.969154	91.74
<b>5246</b>	14:28:02	-7.750681	112.969131	81.69
<b>5247</b>	14:28:03	-7.750646	112.969121	72.85
<b>5248</b>	14:28:05	-7.750569	112.969072	55.17
<b>5249</b>	14:28:06	-7.750536	112.969054	46.02
<b>5250</b>	14:28:07	-7.750499	112.969024	36.88
<b>5251</b>	14:28:08	-7.750471	112.969004	28.04
<b>5252</b>	14:28:09	-7.750429	112.968987	19.51
<b>5253</b>	14:28:10	-7.750386	112.968957	11.58
<b>5254</b>	14:28:12	-7.750363	112.968962	11.58
<b>5255</b>	14:28:13	-7.750361	112.968969	11.58



Berikut adalah visualisasi data dari Tabel A.1



**Gambar A.1.** Visualisasi Data Tampak 2-Dimensi (*Clamped*)



**Gambar A.2.** Visualisasi Data Tampak 3-Dimensi (*Extrude*)





**Gambar A.3.** Visualisasi Data Tampak 3-Dimensi (*Rare*)

## Lampiran B

Pada lampiran berikut ini adalah data pengujian tanpa menggunakan *antenna tracker* dengan balon udara.

**Tabel B.1** Data pengujian tanpa *antenna tracker*

No	Waktu	Latitude	Longitude	Altitude (m)
1	12:47:09	-7.618259	112.846947	4.87
2	12:47:11	-7.618241	112.846912	10.36
3	12:47:12	-7.618189	112.846881	15.24
4	12:47:13	-7.618124	112.846874	21.64
5	12:47:14	-7.618086	112.846876	28.34
6	12:47:15	-7.618098	112.846864	35.05
7	12:47:16	-7.618108	112.846842	41.75
8	12:47:18	-7.618059	112.846731	52.42
9	12:47:19	-7.618011	112.846722	58.82
10	12:47:20	-7.617978	112.846751	66.44
11	12:47:21	-7.617976	112.846772	74.06
12	12:47:22	-7.618001	112.846744	81.99
13	12:47:23	-7.618013	112.846681	89.61
14	12:47:25	-7.617903	112.846591	105.76
15	12:47:26	-7.617856	112.846606	114.90
16	12:47:27	-7.617863	112.846609	123.44
17	12:47:28	-7.617913	112.846589	133.50
18	12:47:29	-7.617944	112.846562	143.56
19	12:47:30	-7.617918	112.846527	152.40
20	12:47:32	-7.617801	112.846509	167.64
21	12:47:33	-7.617796	112.846509	175.26
22	12:47:34	-7.617821	112.846507	182.27
23	12:47:35	-7.617844	112.846472	188.67
24	12:47:36	-7.617834	112.846414	194.15

Lampiran - B2

<b>Tabel B.1(Lanjutan) Data pengujian tanpa <i>antenna tracker</i></b>				
25	12:47:37	-7.617794	112.846382	200.55
26	12:47:38	-7.617754	112.846391	208.48
27	12:47:40	-7.617753	112.846417	226.16
28	12:47:41	-7.617764	112.846386	234.39
29	12:47:42	-7.617766	112.846314	241.09
30	12:47:43	-7.617748	112.846256	247.80
31	12:47:44	-7.617706	112.846254	254.50
32	12:47:45	-7.617658	112.846294	261.51
33	12:47:47	-7.617624	112.846339	277.67
34	12:47:48	-7.617649	112.846299	286.20
35	12:47:49	-7.617649	112.846241	295.35
36	12:47:50	-7.617611	112.846224	303.88
37	12:47:51	-7.617566	112.846254	311.20
38	12:47:52	-7.617544	112.846282	319.12
39	12:47:54	-7.617569	112.846244	332.84
40	12:47:55	-7.617549	112.846189	339.54
41	12:47:56	-7.617508	112.846156	346.86
42	12:47:57	-7.617461	112.846169	354.17
43	12:47:58	-7.617451	112.846204	361.49
44	12:47:59	-7.617468	112.846222	370.63
45	12:48:01	-7.617431	112.846192	385.26
46	12:48:02	-7.617379	112.846204	392.88
47	12:48:03	-7.617354	112.846237	400.81
48	12:48:04	-7.617349	112.846249	409.04
49	12:48:05	-7.617349	112.846226	416.96
50	12:48:06	-7.617341	112.846214	423.97
51	12:48:08	-7.617293	112.846292	436.77
52	12:48:09	-7.617301	112.846316	444.70

<b>Tabel B.1(Lanjutan) Data pengujian tanpa <i>antenna tracker</i></b>				
<b>53</b>	12:48:10	-7.617298	112.846311	452.01
<b>54</b>	12:48:11	-7.617248	112.846287	459.63
<b>55</b>	12:48:12	-7.617169	112.846277	465.42
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
<b>800</b>	13:02:35	-7.625989	112.870942	6292.59
<b>801</b>	13:02:36	-7.625979	112.871016	6299.30
<b>802</b>	13:02:37	-7.626011	112.871072	6306.00
<b>803</b>	13:02:38	-7.626076	112.871124	6313.93
<b>804</b>	13:02:39	-7.626139	112.871206	6321.85
<b>805</b>	13:02:40	-7.626184	112.871337	6329.17
<b>806</b>	13:02:42	-7.626196	112.871642	6341.66
<b>807</b>	13:02:43	-7.626198	112.871717	6348.06
<b>808</b>	13:02:44	-7.626228	112.871752	6355.38
<b>809</b>	13:02:45	-7.626291	112.871814	6363.30
<b>810</b>	13:02:46	-7.626351	112.871941	6370.01
<b>811</b>	13:02:47	-7.626381	112.872117	6376.72
<b>812</b>	13:02:48	-7.626379	112.872271	6383.42
<b>813</b>	13:02:50	-7.626389	112.872381	6396.53
<b>814</b>	13:02:51	-7.626421	112.872419	6404.45
<b>815</b>	13:02:52	-7.626458	112.872512	6412.38
<b>816</b>	13:02:53	-7.626494	112.872659	6420.91
<b>817</b>	13:02:54	-7.626508	112.872806	6429.14
<b>818</b>	13:02:55	-7.626503	112.872906	6437.07
<b>819</b>	13:02:57	-7.626499	112.872952	6452.31
<b>820</b>	13:02:58	-7.626546	112.872986	6460.54
<b>821</b>	13:02:59	-7.626619	112.873067	6467.55
<b>822</b>	13:03:00	-7.626678	112.873182	6475.78



Lampiran - B4

<b>Tabel B.1(Lanjutan) Data pengujian tanpa <i>antenna tracker</i></b>				
<b>823</b>	13:03:01	-7.626701	112.873296	6483.40
<b>824</b>	13:03:02	-7.626689	112.873379	6490.71
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
<b>1615</b>	14:08:05	-7.710704	112.887172	13279.83
<b>1616</b>	14:08:06	-7.710779	112.887282	13263.37
<b>1617</b>	14:08:07	-7.710833	112.887364	13247.52
<b>1618</b>	14:08:08	-7.710896	112.887421	13231.67
<b>1619</b>	14:08:09	-7.710976	112.887504	13215.21
<b>1620</b>	14:08:11	-7.711093	112.887704	13182.90
<b>1621</b>	14:08:12	-7.711148	112.887766	13166.75
<b>1622</b>	14:08:13	-7.711239	112.887822	13150.90
<b>1623</b>	14:08:14	-7.711344	112.887912	13133.83
<b>1624</b>	14:08:15	-7.711428	112.888029	13116.15
<b>1625</b>	14:08:16	-7.711499	112.888116	13099.69
<b>1626</b>	14:08:17	-7.711598	112.888169	13082.93
<b>1627</b>	14:08:19	-7.711819	112.888357	13049.40
<b>1628</b>	14:08:20	-7.711899	112.888454	13033.55
<b>1629</b>	14:08:21	-7.711994	112.888516	13018.31
<b>1630</b>	14:08:22	-7.712099	112.888566	13002.76
<b>1631</b>	14:08:23	-7.712216	112.888641	12987.22
<b>1632</b>	14:08:24	-7.712339	112.888716	12970.76
<b>1633</b>	14:08:28	-7.712821	112.889029	12909.80
<b>1634</b>	14:08:29	-7.712933	112.889102	12894.86
<b>1635</b>	14:08:30	-7.713054	112.889202	12879.32
<b>1636</b>	14:08:34	-7.713509	112.889671	12817.44
<b>1637</b>	14:08:35	-7.713644	112.889797	12802.81
<b>1638</b>	14:08:36	-7.713763	112.889907	12788.18

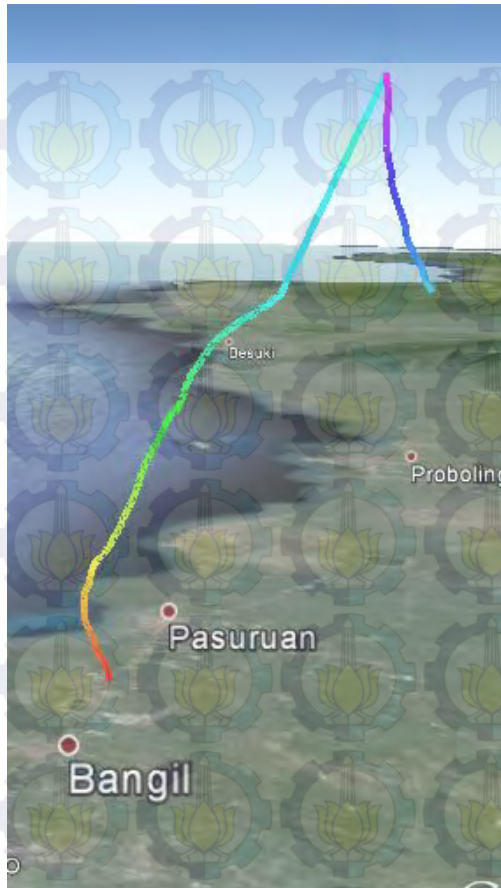
Berikut adalah visualisasi data dari Tabel B.1



**Gambar B.1.** Visualisasi Data Tampak 2-Dimensi (*Clamped*)



**Gambar B.2.** Visualisasi Data Tampak 3-Dimensi (*Extrude*)



**Gambar B.3.** Visualisasi Data Tampak 3-Dimensi (*Rare*)

## Lampiran C

Pada lampiran berikut ini adalah *source code* yang digunakan pada *hardware payload*.

```
#include <SoftwareSerial.h>
#include <TinyGPS.h>
#include <LiquidCrystal.h>

LiquidCrystal lcd(8, 9, 4, 5, 6, 7);    // select the pins used on
the LCD panel
int lcd_key   = 0;
int adc_key_in = 0;

#define btnSELECT 0
#define btnLEFT  1
#define btnUP    2
#define btnDOWN  3
#define btnRIGHT 4
#define btnNONE  5

//SoftwareSerial GPSSerial(3, 2); // RX, TX
TinyGPS gps;
static void print_float(float val, float invalid, int len, int prec);

// turn on GPRMC and GGA
#define          PMTK_SET_NMEA_OUTPUT_RMCGGA
"$PMTK314,0,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0*28"
#define          PMTK_SET_NMEA_UPDATE_1HZ
"$PMTK220,1000*1F"
```



## Lampiran - C2

```
void gpstdump(TinyGPS &gps);  
float printFloat(double f, int digits = 6);  
float mph;  
float alt;  
long lat, lon;  
float flat, flon;  
unsigned long age, date, time, chars;  
byte month, day, hour, minute, second, hundredths;  
int level;  
int year;  
  
boolean senddata = false;  
boolean savehome = false;  
  
void setup()  
{  
  Serial.begin(9600);  
  
  lcd.begin(16, 2);  
  pinMode(10, OUTPUT);  
  digitalWrite(10, 1);  
  
  Serial3.begin(9600);  
  Serial3.println(PMTK_SET_NMEA_OUTPUT_RMCGGA);  
  Serial3.println(PMTK_SET_NMEA_UPDATE_1HZ);  
}  
  
void loop()
```

```

{
  lcd.clear();
  lcd_key = read_LCD_buttons();

  ///////////////////////////////////////////////////

  if (flat <= 0 || flon <= 0){
    lcd.setCursor(2,0);
    lcd.print("GPS NOT FIX");

    senddata = false;
    savehome = false;
  }else{
    if(senddata){
      //Serial.print("E205 ");
      //Serial.print(flat, 6);
      //Serial.print(" ");
      //Serial.print(flon, 6);
      //Serial.print(" ");
      //Serial.println(alt);

      switch (lcd_key){
        case 1:{
          lcd.setCursor(0,0);
          lcd.print("Speed (mph)");
          lcd.setCursor(0,1);
          lcd.print(mph);
          break;
        }
      }
    }
  }
}

```

## Lampiran - C4

```
case 2: {
    lcd.setCursor(0,0);
    lcd.print("Date:  "); lcd.print(static_cast<int>(month));
    lcd.print("/");
    lcd.print(static_cast<int>(day));          lcd.print("/");
    lcd.print(year);
    lcd.setCursor(0,1);
    lcd.print("Time:  ");   lcd.print(static_cast<int>(hour));
    lcd.print(":");
    lcd.print(static_cast<int>(minute));       lcd.print(":");
    lcd.print(static_cast<int>(second));
    lcd.print("."); lcd.print(static_cast<int>(hundredths));
    break;
}
case 3: {
    lcd.setCursor(0,0);
    lcd.print("Altitude (m)");
    lcd.setCursor(0,1);
    lcd.print(alt);
    break;
}
case 4: {
    lcd.setCursor(0,0);
    lcd.print("Level");
    lcd.setCursor(0,1);
    lcd.print(level);
    break;
}
default:
    lcd.setCursor(0,0);
```

```

    lcd.print("Long ");
    lcd.print(flon, 6);
    lcd.setCursor(0,1);
    lcd.print("Lat ");
    lcd.print(flat, 6);
  }
} else {
  if(savehome){
    lcd.setCursor(0,0);
    lcd.print("Save Home Pos.");

    switch (lcd_key){
      case btnSELECT: {
        senddata = true;
        break;
      }
    }
  } else {
    lcd.setCursor(4,0);
    lcd.print("GPS FIX");

    switch (lcd_key){
      case btnSELECT: {
        Serial.print("HOME ");
        Serial.print(flat, 6);
        Serial.print(" ");
        Serial.print(flat, 6);
        Serial.print(" ");
        Serial.println(alt);

```



## Lampiran - C6

```
        savehome = true;
        break;
    }
}

//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

bool newdata = false;
unsigned long start = millis();

// Every 1 seconds we print an update
while (millis() - start < 1000){
    if (Serial3.available()){
        char c = Serial3.read();
        // Serial.print(c); // uncomment to see raw GPS data
        if (gps.encode(c)){
            newdata = true;
            break; // uncomment to print new data immediately!
        }
    }
}

if (newdata){
    gpsdump(gps);
}
```

```
}

```

```
static void print_float(float val, float invalid, int len, int prec)

```

```
{

```

```
    if (val == invalid)

```

```
    {

```

```
        //while (len-- > 1)

```

```
        //Serial.print('*');

```

```
        //Serial.print(' ');

```

```
    }

```

```
    else

```

```
    {

```

```
        //Serial.print(val, prec);

```

```
        int vi = abs((int)val);

```

```
        int flen = prec + (val < 0.0 ? 2 : 1); // . and -

```

```
        flen += vi >= 1000 ? 4 : vi >= 100 ? 3 : vi >= 10 ? 2 : 1;

```

```
        //for (int i=flen; i<len; ++i)

```

```
        //Serial.print(' ');

```

```
    }

```

```
}

```

```
void gpstdump(TinyGPS &gps)

```

```
{

```

```
    //long lat, lon;

```

```
    //float flat, flon;

```

```
    //unsigned long age, date, time, chars;

```

```
    //int year;

```

```
    //byte month, day, hour, minute, second, hundredths;

```

## Lampiran - C8

unsigned short sentences, failed;

```
gps.get_position(&lat, &lon, &age);  
/*Serial.print("Lat/Long(10^-5 deg): ");  
Serial.print(lat);  
Serial.print(", ");  
Serial.print(lon);  
Serial.print(" Fix age: ");  
Serial.print(age);  
Serial.println("ms.");*/
```

// On Arduino, GPS characters may be lost during lengthy  
Serial.print()

// On Teensy, Serial prints to USB, which has large output  
buffering and

// runs very fast, so it's not necessary to worry about missing  
4800

// baud GPS characters.

```
gps.f_get_position(&flat, &flon, &age);  
print_float(flat, TinyGPS::GPS_INVALID_F_ANGLE, 10, 6);  
print_float(flon, TinyGPS::GPS_INVALID_F_ANGLE, 11, 6);
```

```
/*Serial.print("Lat/Long(float): ");  
printFloat(flat, 5);  
Serial.print(", ");  
printFloat(flon, 5);  
Serial.print(" Fix age: ");  
Serial.print(age);  
Serial.println("ms.");*/
```

```

gps.get_datetime(&date, &time, &age);
/*Serial.print("Date(ddmmyy): ");
Serial.print(date);
Serial.print(" Time(hhmmsscc): ");
Serial.print(time);
Serial.print(" Fix age: ");
Serial.print(age);
Serial.println("ms.");*/

```

```

gps.crack_datetime(&year, &month, &day, &hour, &minute,
&second, &hundredths, &age);
/*Serial.print("Date: ");
Serial.print(static_cast<int>(month));
Serial.print("/");
Serial.print(static_cast<int>(day));
Serial.print("/");
Serial.print(year);
Serial.print(" Time: ");
Serial.print(static_cast<int>(hour));
Serial.print(":");
Serial.print(static_cast<int>(minute));
Serial.print(":");
Serial.print(static_cast<int>(second));
Serial.print(".");
Serial.print(static_cast<int>(hundredths));
Serial.print(" Fix age: ");
Serial.print(age);
Serial.println("ms.");*/

```



```
/*Serial.print("Alt(cm): ");
Serial.print(gps.altitude());
Serial.print(" Course(10^-2 deg): ");
Serial.print(gps.course());
Serial.print(" Speed(10^-2 knots): ");
Serial.println(gps.speed());
Serial.print("Alt(float): ");
printFloat(gps.f_altitude());
Serial.print(" Course(float): ");
printFloat(gps.f_course());
Serial.println();
Serial.print("Speed(knots): ");
printFloat(gps.f_speed_knots());
Serial.print(" (mph): ");
printFloat(gps.f_speed_mph());
Serial.print(" (mps): ");
printFloat(gps.f_speed_mps());
Serial.print(" (kmph): ");
printFloat(gps.f_speed_kmph());
Serial.println();*/

mph=printFloat(gps.f_speed_mph(), 2);
alt=printFloat(gps.f_altitude(), 2);

gps.stats(&chars, &sentences, &failed);
/*Serial.print("Stats: characters: ");
Serial.print(chars);
Serial.print(" sentences: ");
Serial.print(sentences);
```

```
Serial.print(" failed checksum: ");
Serial.println(failed);*/
```

```
Serial.print("E205 ");
Serial.print(flat, 6);
Serial.print(" ");
Serial.print(flon, 6);
Serial.print(" ");
Serial.println(alt);
}
```

```
float printFloat(double number, int digits)
```

```
{
    // Handle negative numbers
    if (number < 0.0) {
        //Serial.print('-');
        number = -number;
    }
}
```

```
// Round correctly so that print(1.999, 2) prints as "2.00"
```

```
double rounding = 0.5;
```

```
for (uint8_t i=0; i<digits; ++i)
```

```
    rounding /= 10.0;
```

```
number += rounding;
```

```
// Extract the integer part of the number and print it
```

```
unsigned long int_part = (unsigned long)number;
```

```
double remainder = number - (double)int_part;
```

```
//Serial.print(int_part);
```

## Lampiran - C12

```
// Print the decimal point, but only if there are digits beyond
if (digits > 0)
    //Serial.print(".");

// Extract digits from the remainder one at a time
while (digits-- > 0) {
    remainder *= 10.0;
    int toPrint = int(remainder);
    //Serial.print(toPrint);
    remainder -= toPrint;
}
}

int read_LCD_buttons() {    // read the buttons
    adc_key_in = analogRead(0);    // read the value from the sensor

    if (adc_key_in > 1000) return btnNONE;
    if (adc_key_in < 50)  return btnRIGHT;
    if (adc_key_in < 250) return btnUP;
    if (adc_key_in < 450) return btnDOWN;
    if (adc_key_in < 650) return btnLEFT;
    if (adc_key_in < 850) return btnSELECT;

    return btnNONE;    // when all others fail, return this.
}
```

## Lampiran D

Pada lampiran berikut ini adalah *source code* yang digunakan pada *hardware tracker* .

```
#include <Servo.h>

const char EOM = '#';

Servo pan; // pan / yaw / azimuth
Servo tilt; // tilt / pitch / elevasi

long lastPan = 823L;
long lastTilt = 0L;

int minPAN = 600;
int maxPAN = 823;
int minTILT = 620;
int maxTILT = 2520;

double panVal;
double tiltVal;
double interpVal;
int tiltDelay = 15;
int dir = 1;
int pos = 0;

String cmd = "";
String panStr = "";
String tiltStr = "";
```



## Lampiran - D2

```
double data, pan_filter, tilt_filter;
float p_filter = 0.03; // 0.01 to 1.0
float t_filter = 0.03; // 0.01 to 1.0
int index_azi, index_elev;

double mapping(double x, double in_min, double in_max, double
out_min, double out_max)
{
    return (x - in_min) * (out_max - out_min) / (in_max - in_min) +
out_min;
}

void setup()
{
    pan.attach(9, minPAN, maxPAN);
    tilt.attach(10, minTILT, maxTILT);

    Serial.begin(38400);

    pan.writeMicroseconds(lastPan);
    tilt.write(lastTilt);

    delay(2000);
    Serial.println("38400");
}

void loop()
{
```

```

char ch;
if (Serial.available() > 0)
{
    ch = Serial.read();
    if (ch == EOM)
    {
        index_azi = cmd.indexOf("A");
        index_elev = cmd.indexOf("E");

        if (index_azi != -1 && index_elev != -1)
        {
            panStr = cmd.substring(index_azi+1, index_elev-1);
            if (panStr != "")
            {
                panVal = panStr.toInt();
                //data = mapping((panVal / 100), 0, 359, minPAN,
                maxPAN);

                if((panVal / 100) >= 0 && (panVal / 100) <= 90){
                    data = mapping((panVal / 100), 0, 90, minPAN, 660);
                }else if((panVal / 100) > 90 && (panVal / 100) <= 180){
                    data = mapping((panVal / 100), 91, 180, 660, 718);
                }else if((panVal / 100) > 180 && (panVal / 100) <= 270){
                    data = mapping((panVal / 100), 181, 270, 718, 770);
                }else if((panVal / 100) > 270 && (panVal / 100) <= 360){
                    data = mapping((panVal / 100), 271, 360, 770, maxPAN);
                }

                /*pan_filter = pan_filter * (1.0-p_filter) + data * p_filter;
                pan.writeMicroseconds(pan_filter);

```

## Lampiran - D4

```
lastPan = pan_filter;*/

pan.writeMicroseconds(data);
lastPan = data;
//Serial.print("yaw "); Serial.print((panVal / 100));
Serial.print(", microsecond "); Serial.println(data);
}

/*double data, pan_filter, tilt_filter;
float p_filter = 0.03; // 0.01 to 1.0
float t_filter = 0.05; // 0.01 to 1.0*/

tiltStr = cmd.substring(index_elev+1);
if (tiltStr != "")
{
    tiltVal = tiltStr.toInt();
    data = tiltVal / 100;

    /*tilt_filter = tilt_filter * (1.0-t_filter) + data * t_filter;
    tilt.write(tilt_filter);
    lastTilt = tilt_filter;*/

    tilt.write(data);
    lastTilt = data;

    data = mapping(lastTilt, 0, 180, minTILT, maxTILT);

    //Serial.print("pitch "); Serial.print(lastTilt); Serial.print(",
microsecond "); Serial.println(data);
}
```

```
//Serial.print(cmd); Serial.print(" ");Serial.print(panVal);
Serial.print(" "); Serial.println(tiltVal);
}

panStr = "";
tiltStr = "";
cmd = "";
} else {
    cmd += ch;
}
}
```





## Lampiran E

Pada lampiran berikut ini adalah *source code* yang digunakan pada *downlink* .

```
Imports System
```

```
Imports System.Data.OleDb
```

```
Imports System.ComponentModel
```

```
Imports System.Threading
```

```
Imports System.IO
```

```
Imports Excel = Microsoft.Office.Interop.Excel
```

```
Imports ZedGraph
```

```
Public Class Downlink_Main
```

```
    Dim SerialList As New Collection
```

```
    Dim SerialPayload As String = "null"
```

```
    Dim SerialAntenna As String = "null"
```

```
    Dim _SerialPort As String = ""
```

```
    Dim Buffer As String = ""
```

```
    Dim _datasend As String
```

```
    Dim APP As New Excel.Application
```

```
    Dim WS1, WS2, WS3, WS4 As Excel.Worksheet
```

```
    Dim WB As Excel.Workbook
```

```
#Region " HAVERSINE FORMULA "
```

```
    Private Sub CalcDGVToANT()
```

```
        Dim Transform As New tmapsapi.NMEA_Transform
```

```
        Dim oTarget As tmapsapi.NMEA_Transform.structTarget =  
        Nothing
```

```
oTarget =  
Transform.GetCourseAndDistance(CType(txt_Lat1.Text,  
Double), CType(txt_Lon1.Text, Double), CType(txt_Alt1.Text,  
Double), CType(txt_Lat2.Text, Double), CType(txt_Lon2.Text,  
Double), CType(txt_Alt2.Text, Double))  
txt_Dis.Text = Math.Round(oTarget.Distance, 2) ' .ToString  
' & " Meters"  
txt_Azi.Text = Math.Round(oTarget.Course, 2).ToString  
  
If in_KM Then  
    Dis_Miles = (Cdbl(txt_Dis.Text) / 1000) * 0.6214  
Else  
    Dis_Miles = Cdbl(txt_Dis.Text) / 5280  
End If  
  
txt_Elev.Text = (CalElevation(((Cdbl(txt_Alt1.Text)) *  
0.000621371), ((Cdbl(txt_Alt2.Text)) * 0.000621371),  
Dis_Miles)).ToString("F2")  
  
'If txt_Lat2.Text <> b_lat And txt_Lon2.Text <> b_lon And  
txt_Alt2.Text <> b_alt And d_temp <> b_temp And d_press <>  
b_press And d_humid <> b_humid Then  
    Dim str(Me.dgvData2.ColumnCount - 1) As String  
  
    str(lat.Index) = txt_Lat2.Text  
    str(lon.Index) = txt_Lon2.Text  
    str(alt.Index) = txt_Alt2.Text  
  
    str(azi2.Index) = txt_Azi.Text ' & "°" ' PAN  
    str(dis2.Index) = txt_Dis.Text
```

```
str(elev2.Index) = txt_Elev.Text '& "°"      ' TILT
```

```
str(temp.Index) = d_temp
```

```
str(press.Index) = d_press
```

```
str(humid.Index) = d_humid
```

```
dgvData2.FirstDisplayedScrollingRowIndex =  
dgvData2.RowCount - 1
```

```
dgvData2.Rows.Add(str)
```

```
'b_lat = txt_Lat2.Text
```

```
'b_lon = txt_Lon2.Text
```

```
'b_alt = txt_Alt2.Text
```

```
'b_temp = d_temp
```

```
'b_press = d_press
```

```
'b_humid = d_humid
```

```
_datasend = "A" & (CType(txt_Azi.Text, Double) * 100) & "  
E" & (CType(txt_Elev.Text, Double) * 100) & "#"  
SerialportX_DataWrite()
```

```
'Me.Text = _datasend
```

```
'End If
```

```
End Sub
```

```
Private Sub HaversineFormula()
```

```
Try
```

```
LatRad(1) = CDbl(txt_Lat1.Text) * Math.PI / 180.0
```



#### Lampiran - E4

```
LonRad(1) = CDb1(txt_Lon1.Text) * Math.PI / 180.0
```

```
LatRad(2) = CDb1(txt_Lat2.Text) * Math.PI / 180.0
```

```
LonRad(2) = CDb1(txt_Lon2.Text) * Math.PI / 180.0
```

```
Catch ex As Exception
```

```
    MessageBox.Show("Incorrect data entry!")
```

```
Exit Sub
```

```
End Try
```

```
txt_Dis.Text = (CalDistance(LatRad(1), LonRad(1),  
LatRad(2), LonRad(2), in_KM)).ToString("F4")
```

```
txt_Azi.Text = (CalAzimuth(LatRad(1), LonRad(1),  
LatRad(2), LonRad(2))).ToString("F2")
```

```
If in_KM Then
```

```
    Dis_Miles = CDb1(txt_Dis.Text) * 0.6214
```

```
Else
```

```
    Dis_Miles = CDb1(txt_Dis.Text) / 5280
```

```
End If
```

```
txt_Elev.Text = (CalElevation(((CDb1(txt_Alt1.Text)) *  
0.000621371), ((CDb1(txt_Alt2.Text)) * 0.000621371),  
Dis_Miles)).ToString("F2")
```

```
Dim str(Me.dgvData2.ColumnCount - 1) As String
```

```
str(lat.Index) = txt_Lat2.Text
```

```
str(lon.Index) = txt_Lon2.Text
```

```
str(alt.Index) = txt_Alt2.Text
```

```
str(azi2.Index) = txt_Azi.Text '& "°" ' PAN
```

```

str(dis2.Index) = txt_Dis.Text
str(elev2.Index) = txt_Elev.Text & "°"      ' TILT

str(temp.Index) = d_temp
str(press.Index) = d_press
str(humid.Index) = d_humid

    _datasend = "A" & (CType(txt_Azi.Text, Double) * 100) & "
E" & (CType(txt_Elev.Text, Double) * 100) & "#"
    SerialportX_DataWrite()

    dgvData2.FirstDisplayedScrollingRowIndex =
dgvData2.RowCount - 1
    dgvData2.Rows.Add(str)

End Sub
#End Region

#Region " COMMUNICATION "

#Region " COMMAND "

    Private Sub btn_connect_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btn_connect.Click
        If cmb_PayPort.Text <> "" And cmb_AntPort.Text <> ""
Then
            Dim SerialPort As System.IO.Ports.SerialPort
            SerialPort = New System.IO.Ports.SerialPort
            Try
                If SerialPayload = "null" Then
                    SerialPort = New System.IO.Ports.SerialPort

```

With SerialPort

```
.PortName = cmb_PayPort.SelectedItem
.BaudRate = cmb_PayBaud.SelectedItem
.Parity = IO.Ports.Parity.None
.StopBits = IO.Ports.StopBits.One
.DataBits = 8
.Open()
.DtrEnable = True
rtx_RawData.Clear()

AddHandler SerialPort.DataReceived, AddressOf
SerialportX_DataReceived
SerialList.Add(SerialPort, SerialPort.PortName)
SerialPayload =
cmb_PayPort.Items(cmb_PayPort.SelectedIndex)
End With
```

SerialPort = New System.IO.Ports.SerialPort

With SerialPort

```
.PortName = cmb_AntPort.SelectedItem
.BaudRate = cmb_AntBaud.SelectedItem
.Parity = IO.Ports.Parity.None
.StopBits = IO.Ports.StopBits.One
.DataBits = 8
.Open()
AddHandler SerialPort.DataReceived, AddressOf
SerialportX_DataReceived
SerialList.Add(SerialPort, SerialPort.PortName)
SerialAntenna =
cmb_AntPort.Items(cmb_AntPort.SelectedIndex)
```

End With

btn\_connect.Text = "Disconnect"

cmb\_PayPort.Enabled = False

cmb\_AntPort.Enabled = False

Else

SerialPort = SerialList.Item(SerialPayload)

If SerialPort.IsOpen Then

SerialPort.Close()

RemoveHandler SerialPort.DataReceived,  
AddressOf SerialportX\_DataReceived

End If

SerialList.Remove(SerialPayload)

SerialPayload = "null"

SerialPort = SerialList.Item(SerialAntenna)

If SerialPort.IsOpen Then

SerialPort.Close()

RemoveHandler SerialPort.DataReceived,  
AddressOf SerialportX\_DataReceived

End If

SerialList.Remove(SerialAntenna)

SerialAntenna = "null"

btn\_connect.Text = "Connect"

cmb\_PayPort.Enabled = True

cmb\_AntPort.Enabled = True

End If

Catch ex As Exception

MsgBox(ex.Message)



Finally

```
End Try
Else
    MsgBox("Select Communication Port")
End If
End Sub
#End Region

#Region " PAYLOAD "
Private Sub cmb_PayPort_DropDown(ByVal sender As Object,
ByVal e As System.EventArgs) Handles cmb_PayPort.DropDown
    cmb_PayPort.Items.Clear()
    For i As Integer = 0 To
My.Computer.Ports.SerialPortNames.Count - 1

cmb_PayPort.Items.Add(My.Computer.Ports.SerialPortNames(i))
Next
End Sub

Private Sub cmb_PayPort_SelectedIndexChanged(ByVal
sender As System.Object, ByVal e As System.EventArgs) Handles
cmb_PayPort.SelectedIndexChanged
    If SerialPort1.IsOpen = False Then
        SerialPort1.PortName = cmb_PayPort.Text 'pop a
message box to user if he is changing ports
    Else
        'without disconnecting first.
        MsgBox("Valid only if port is Closed", vbCritical)
    End If
End Sub
```

```
Private Sub cmb_PayBaud_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmb_PayBaud.SelectedIndexChanged
```

```
    If SerialPort1.IsOpen = False Then
```

```
        SerialPort1.BaudRate = cmb_PayBaud.Text 'pop a message box to user if he is changing baud rate
```

```
    Else 'without disconnecting first.
```

```
        MsgBox("Valid only if port is Closed", vbCritical)
```

```
    End If
```

```
End Sub
```

```
#End Region
```

```
#Region " TRACKER "
```

```
Private Sub cmb_AntPort_DropDown(ByVal sender As Object, ByVal e As System.EventArgs) Handles cmb_AntPort.DropDown
```

```
    cmb_AntPort.Items.Clear()
```

```
    For i As Integer = 0 To My.Computer.Ports.SerialPortNames.Count - 1
```

```
        cmb_AntPort.Items.Add(My.Computer.Ports.SerialPortNames(i))
```

```
    Next
```

```
End Sub
```

```
Private Sub cmb_AntPort_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmb_AntPort.SelectedIndexChanged
```

```
    If SerialPort1.IsOpen = False Then
```

```
        SerialPort1.PortName = cmb_AntPort.Text 'pop a message box to user if he is changing ports
```

```
    Else 'without disconnecting first.
```

```
        MsgBox("Valid only if port is Closed", vbCritical)
```

## Lampiran - E10

End If

End Sub

```
Private Sub cmb_AntBaud_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles cmb_AntBaud.SelectedIndexChanged
```

```
    If SerialPort1.IsOpen = False Then
```

```
        SerialPort1.BaudRate = cmb_AntBaud.Text 'pop a message box to user if he is changing baud rate
```

```
    Else 'without disconnecting first.
```

```
        MsgBox("Valid only if port is Closed", vbCritical)
```

```
    End If
```

```
End Sub
```

```
#End Region
```

```
#Region " WRITE "
```

```
Private Sub SerialportX_DataWrite()
```

```
    Dim SerialPort As System.IO.Ports.SerialPort = SerialList(cmb_AntPort.SelectedItem)
```

```
    'rtx_RawData.Text = ""
```

```
    SerialPort.Write(_datasend)
```

```
End Sub
```

```
#End Region
```

```
#Region " READ "
```

```
Private Sub SerialportX_DataReceived(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
Try
```

```
    Dim SerialPort As System.IO.Ports.SerialPort = sender
```

```
    Buffer = SerialPort.ReadLine
```

```
    _SerialPort = SerialPort.PortName
```

```

        Me.BeginInvoke(New EventHandler(AddressOf updrtb))
    Catch ex As Exception

    End Try
End Sub

Sub updrtb(ByVal sender As Object, ByVal e As
System.EventArgs)

    If _SerialPort = cmb_PayPort.SelectedItem Then
        If rtx_RawData.InvokeRequired Then

            End If
            rtx_RawData.AppendText(Buffer)
            rtx_RawData.ScrollToCaret()

            If Microsoft.VisualBasic.Left(Buffer, 4) = "HOME" And
Buffer.Length >= 25 Or Microsoft.VisualBasic.Left(Buffer, 4) =
"E205" And Buffer.Length >= 35 Then
                ListBox1.Items.Add(Buffer)
                ListBox1.SelectedIndex = ListBox1.Items.Count - 1
                Parsing_Data(ListBox1.SelectedItem)
            End If
        End If

        If _SerialPort = cmb_AntPort.SelectedItem Then
            'RichTextBox2.AppendText(Buffer)
            'RichTextBox2.ScrollToCaret()
        End If
    End Sub
#End Region

```



```
#Region " PARSING "
```

```
Private Sub Parsing_Data(ByVal _buf As String)
```

```
    Dim str As String
```

```
    Dim strArr() As String
```

```
    Dim count As Integer
```

```
    str = _buf
```

```
    strArr = str.Split(New Char(-1) {},  
StringSplitOptions.RemoveEmptyEntries)
```

```
    If Microsoft.VisualBasic.Left(_buf, 4) = "HOME" Then ' And  
_buf.Length >= 25 Then
```

```
        For count = 0 To strArr.Length - 1
```

```
            txt_Lat1.Text = strArr(1)
```

```
            txt_Lon1.Text = strArr(2)
```

```
            txt_Alt1.Text = strArr(3)
```

```
        Next
```

```
    ElseIf Microsoft.VisualBasic.Left(_buf, 4) = "E205" Then '  
And _buf.Length >= 35 Then
```

```
        For count = 0 To strArr.Length - 1
```

```
            txt_Lat2.Text = strArr(1)
```

```
            txt_Lon2.Text = strArr(2)
```

```
            txt_Alt2.Text = strArr(3)
```

```
            d_temp = strArr(4)
```

```
            d_press = strArr(5)
```

```
            d_humid = strArr(6)
```

```
            "HaversineFormula()
```

```
            'CalcDGVTtoANT()
```

```

        _Draw(zg_press, d_temp, CType(d_press, Double))
        _Draw(zg_humid, d_temp, CType(d_humid, Double))
        _Draw(zg_alti, d_temp, CType(txt_Alt2.Text, Double))
    Next
End If
End Sub
#End Region

#End Region

#Region " BACKUP DATA "
    Private Sub btn_backup_Click(ByVal sender As System.Object,
        ByVal e As System.EventArgs) Handles btn_backup.Click
        If dgvData2.Rows.Count > 1 Then
            SaveDataGridView(dgvData2)
        Else
            MsgBox("Empty")
        End If
    End Sub
#End Region

#Region " GRAPHIC "
    Private Sub _AltiLoad(ByVal zgc As ZedGraphControl)
        Dim myPane As GraphPane = zgc.GraphPane

        myPane.XAxis.Scale.Min = -40 'minimum value of x-axis
        myPane.XAxis.Scale.Max = 40 'maximum value of x-axis
        myPane.YAxis.Scale.Min = 830 'minimum value of y-axis
        myPane.YAxis.Scale.Max = 101235 'maximum value of y-
axis

```

```
myPane.YAxis.Scale.FontSpec.FontColor = Color.Green  
myPane.YAxis.Title.FontSpec.FontColor = Color.Green  
myPane.Chart.Fill = New Fill(Color.White,  
Color.LightGoldenrodYellow, 45.0F)
```

```
myPane.Title.Text = "Altitude VS Temperature"  
myPane.XAxis.Title.Text = "Temperature"  
myPane.YAxis.Title.Text = "Altitude"
```

```
Dim myCurve As LineItem = myPane.AddCurve("X", New  
ZedGraph.PointPairList(), Color.DarkBlue, SymbolType.None)  
myCurve.Line.IsSmooth = True  
myCurve.Line.SmoothTension = 0.1F
```

```
'myPane.AddCurve("X", New PointPairList, Color.Blue,  
SymbolType.None)
```

```
'myPane.XAxis.IsVisible = False  
zgc.AxisChange()
```

```
End Sub
```

```
Private Sub _PressLoad(ByVal zgc As ZedGraphControl)
```

```
Dim myPane As GraphPane = zgc.GraphPane
```

```
myPane.XAxis.Scale.Min = -40 'minimum value of x-axis  
myPane.XAxis.Scale.Max = 40 'maximum value of x-axis  
myPane.YAxis.Scale.Min = 8 'minimum value of y-axis  
myPane.YAxis.Scale.Max = 989 'maximum value of y-axis
```

```
myPane.YAxis.Scale.FontSpec.FontColor = Color.Green
```

```

myPane.YAxis.Title.FontSpec.FontColor = Color.Green
myPane.Chart.Fill = New Fill(Color.White,
Color.LightGoldenrodYellow, 45.0F)

myPane.Title.Text = "Pressure VS Temperature"
myPane.XAxis.Title.Text = "Temperature"
myPane.YAxis.Title.Text = "Pressure"

Dim myCurve As LineItem = myPane.AddCurve("X", New
ZedGraph.PointPairList(), Color.Red, SymbolType.None)
myCurve.Line.IsSmooth = True
myCurve.Line.SmoothTension = 0.1F
'myPane.AddCurve("Y", New PointPairList, Color.Red,
SymbolType.None)
'myPane.XAxis.IsVisible = False
zgc.AxisChange()
End Sub

Private Sub _HumidLoad(ByVal zgc As ZedGraphControl)
Dim myPane As GraphPane = zgc.GraphPane

myPane.XAxis.Scale.Min = -40 'minimum value of x-axis
myPane.XAxis.Scale.Max = 40 'maximum value of x-axis
myPane.YAxis.Scale.Min = 0 'minimum value of y-axis
myPane.YAxis.Scale.Max = 50 'maximum value of y-axis

myPane.YAxis.Scale.FontSpec.FontColor = Color.Green
myPane.YAxis.Title.FontSpec.FontColor = Color.Green
myPane.Chart.Fill = New Fill(Color.White,
Color.LightGoldenrodYellow, 45.0F)

```



```
myPane.Title.Text = "Humidity VS Temperature"  
myPane.XAxis.Title.Text = "Temperature"  
myPane.YAxis.Title.Text = "Humidity"
```

```
Dim myCurve As LineItem = myPane.AddCurve("Z", New  
ZedGraph.PointPairList(), Color.Green, SymbolType.None)  
myCurve.Line.IsSmooth = True  
myCurve.Line.SmoothTension = 0.1F  
'myPane.AddCurve("Z", New PointPairList, Color.Green,  
SymbolType.None)  
'myPane.XAxis.IsVisible = False  
zgc.AxisChange()  
End Sub
```

```
Private Sub _Draw(ByVal zgc As ZedGraphControl, ByVal  
_TempVal As Double, ByVal _YVal As Double)  
Dim _gp As GraphPane = zgc.GraphPane  
Dim _X As PointPairList =  
zgc.GraphPane.CurveList(0).Points  
_X.Add(_TempVal, _YVal)  
_gp.XAxis.Scale.MaxAuto = True  
_gp.XAxis.Scale.MinAuto = True  
_gp.YAxis.Scale.MaxAuto = True  
_gp.YAxis.Scale.MinAuto = True  
  
zgc.IsShowPointValues = True
```

```

_gp.AxisChange()
zgc.Invalidate()
End Sub

Private Sub _Clear(ByVal zgc As ZedGraphControl)
    Dim _X As PointPairList =
zgc.GraphPane.CurveList(0).Points

    _X.Clear()
    zgc.Invalidate()

    _PressLoad(zg_press)
    _HumidLoad(zg_humid)
    _AltiLoad(zg_alti)

    dgvData2.Rows.Clear()
    ListBox1.Items.Clear()
End Sub
#End Region

#Region " EXCEL DATA "
Private Sub btn_runExcel_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
btn_runExcel.Click
    If cmb_AntPort.Enabled = False Then
        If cmb_ExcelData.Text <> String.Empty Then
            If btn_runExcel.Text = "Run" Then
                _row = 3
                _row2 = 3
            End If
        End If
    End If
End Sub

```

## Lampiran - E18

```
_row3 = 3
_row4 = 3

_Clear(zg_press)
_Clear(zg_humid)
_Clear(zg_alti)

APP = New Excel.Application

If cmb_ExcelData.Text = "FLIGHT TEST" Then
    WB =
APP.Workbooks.Open(Application.StartupPath &
"\Data\FlyHigh.xlsx")

    WS1 = WB.Worksheets("GPS")
    LR1 = WS1.Cells(WS1.Rows.Count,
1).End(Excel.XlDirection.xlUp).Row
    'LC1 = WS1.Cells(1,
WS1.Columns.Count).End(Excel.XlDirection.xlToLeft).Column
    'd_time = ElapsedTime(WS1.Cells(2, 1).Value)
    d_lat = (WS1.Cells(2, 2).Value)
    d_lon = (WS1.Cells(2, 3).Value)
    d_alt = (WS1.Cells(2, 4).Value)

    WS2 = WB.Worksheets("Temperature")
    'LR2 = WS2.Cells(WS2.Rows.Count,
1).End(Excel.XlDirection.xlUp).Row
    'LC2 = WS2.Cells(1,
WS2.Columns.Count).End(Excel.XlDirection.xlToLeft).Column
    d_temp = (WS2.Cells(2, 2).Value)
```

```

WS3 = WB.Worksheets("Pressure")
'LR3 = WS3.Cells(WS3.Rows.Count,
1).End(Excel.XlDirection.xlUp).Row
'LC3 = WS3.Cells(1,
WS3.Columns.Count).End(Excel.XlDirection.xlToLeft).Column
d_press = (WS3.Cells(2, 2).Value)

```

```

WS4 = WB.Worksheets("Humidity")
'LR4 = WS4.Cells(WS4.Rows.Count,
1).End(Excel.XlDirection.xlUp).Row
'LC4 = WS4.Cells(1,
WS4.Columns.Count).End(Excel.XlDirection.xlToLeft).Column
d_humid = (WS4.Cells(2, 2).Value)

```

```

'dgvData.Visible = False
dgvData2.Visible = True

```

```

Dim str(Me.dgvData2.ColumnCount - 1) As String
txt_Lat1.Text = d_lat
txt_Lon1.Text = d_lon
txt_Alt1.Text = d_alt

str(lat.Index) = d_lat
str(lon.Index) = d_lon

```



```
str(alt.Index) = d_alt
```

```
str(azi2.Index) = 0
```

```
str(dis2.Index) = 0
```

```
str(elev2.Index) = 0
```

```
str(temp.Index) = d_temp
```

```
str(press.Index) = d_press
```

```
str(humid.Index) = d_humid
```

```
dgvData2.FirstDisplayedScrollingRowIndex =  
dgvData2.RowCount - 1
```

```
dgvData2.Rows.Add(str)
```

```
tmr_runExcel.Interval = 2000
```

```
ListBox1.Items.Add("HOME " & d_lat & " " &  
d_lon & " " & d_alt)
```

```
End If
```

```
cmb_ExcelData.Enabled = False
```

```
tmr_runExcel.Enabled = True
```

```
btn_runExcel.Text = "Stop"
```

```
Else
```

```
Try
```

```
cmb_ExcelData.Enabled = True
```

```
tmr_runExcel.Enabled = False
```

```
btn_runExcel.Text = "Run"
```

```
WB.Save()
```

```
WB.Close()
```

```
APP.Quit()
```

```
If cmb_ExcelData.Text = "FLIGHT TEST" Then
```

```
System.Runtime.InteropServices.Marshal.ReleaseComObject(WS  
1)
```

```
System.Runtime.InteropServices.Marshal.ReleaseComObject(WS  
2)
```

```
System.Runtime.InteropServices.Marshal.ReleaseComObject(WS  
3)
```

```
System.Runtime.InteropServices.Marshal.ReleaseComObject(WS  
4)
```

```
System.Runtime.InteropServices.Marshal.ReleaseComObject(W  
B)
```

```
System.Runtime.InteropServices.Marshal.ReleaseComObject(AP  
P)
```

```
WS1 = Nothing
```

```
WS2 = Nothing
```

```
WS3 = Nothing
```

```
WS4 = Nothing
```

```
WB = Nothing
```

```
APP = Nothing
```

End If

```
GC.GetTotalMemory(False)
GC.Collect()
GC.WaitForPendingFinalizers()
GC.Collect()
GC.GetTotalMemory(True)
```

Catch ex As Exception

Finally

If APP IsNot Nothing Then

If cmb\_ExcelData.Text = "FLIGHT TEST" Then

System.Runtime.InteropServices.Marshal.ReleaseComObject(WS  
1)

System.Runtime.InteropServices.Marshal.ReleaseComObject(WS  
2)

System.Runtime.InteropServices.Marshal.ReleaseComObject(WS  
3)

System.Runtime.InteropServices.Marshal.ReleaseComObject(WS  
4)

System.Runtime.InteropServices.Marshal.ReleaseComObject(W  
B)

System.Runtime.InteropServices.Marshal.ReleaseComObject(AP  
P)

```

APP = Nothing
End If
End If
End Try
End If
Else
MsgBox("Choose Data")
End If
Else
MsgBox("Antenna Not Connected")
End If
End Sub

Private Sub tmr_runExcel_Tick(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
tmr_runExcel.Tick
If cmb_ExcelData.Text = "FLIGHT TEST" Then
If _row <= LR1 Then
'd_time = ElapsedTime(WS1.Cells(_row, 1).Value) '1
'd_time2 = ElapsedTime(WS2.Cells(_row2, 1).Value)
'2
'd_time3 = ElapsedTime(WS3.Cells(_row3, 1).Value)
'2
'd_time4 = ElapsedTime(WS4.Cells(_row4, 1).Value)
'2

txt_Lat2.Text = WS1.Cells(_row, 2).Value
txt_Lon2.Text = WS1.Cells(_row, 3).Value
txt_Alt2.Text = WS1.Cells(_row, 4).Value

If d_time2 = d_time Then

```



## Lampiran - E24

```
d_temp = (WS2.Cells(_row, 2).Value)
'_row2 += 1
'End If

'If d_time3 = d_time Then
d_press = (WS3.Cells(_row, 2).Value)
'_row3 += 1
'End If

'If d_time4 = d_time Then
d_humid = (WS4.Cells(_row, 2).Value)
'_row4 += 1
'End If

ListBox1.Items.Add("E205 " & d_lat & " " & d_lon & "
" & d_alt & " " & d_temp & " " & d_press & " " & d_humid)

CalcDGVToANT()

_row += 1

_Draw(zg_press, d_temp, CType(d_press, Double))
_Draw(zg_humid, d_temp, CType(d_humid, Double))
_Draw(zg_alti, d_temp, CType(txt_Alt2.Text, Double))

End If
End If
End Sub
#End Region
```

```
Private Sub Downlink_Main_Load(ByVal sender As  
System.Object, ByVal e As System.EventArgs) Handles  
MyBase.Load  
    _PressLoad(zg_press)  
    _HumidLoad(zg_humid)  
    _AltiLoad(zg_alti)  
End Sub  
End Class
```



## Lampiran F

Pada lampiran berikut ini adalah *source code* yang digunakan pada *module downlink* .

### Module Downlink\_Global

Public Radius\_KM As Double = 6371

Public Radius\_FT As Double = 20902230.971128609

Public Radius\_Miles As Double = 3958.756

Public Dis\_Miles As Double

Public in\_KM As Boolean = True

Public LatRad(2) As Double

Public LonRad(2) As Double

Public d\_lat, d\_lon, d\_alt, d\_temp, d\_press, d\_humid As String

Public b\_lat, b\_lon, b\_alt, b\_temp, b\_press, b\_humid As String

Public LC1, LR1 As Long

Public \_row, \_row2, \_row3, \_row4 As Integer

Public d\_time, d\_time2, d\_time3, d\_time4 As String

Public Function CalDistance(ByVal Lat1 As Double, ByVal Lon1 As Double, ByVal Lat2 As Double, ByVal Lon2 As Double, ByRef KM As Boolean) As Double

'Haversine Formula

Dim dLat As Double = Lat2 - Lat1

Dim dLong As Double = Lon2 - Lon1



## Lampiran - F2

```
Dim a As Double = Math.Sin(dLat / 2.0) * Math.Sin(dLat /  
2.0) + _
```

```
Math.Cos(Lat1) * Math.Cos(Lat2) * Math.Sin(dLong / 2.0)  
* Math.Sin(dLong / 2.0)
```

```
Dim c As Double = 2.0 * Math.Atan2(Math.Sqrt(a),  
Math.Sqrt(1.0 - a))
```

```
If KM Then
```

```
CalDistance = Radius_KM * c
```

```
Else
```

```
CalDistance = Radius_FT * c
```

```
End If
```

```
'This must be a Spherical Law of Cosines
```

```
' CalDistance = Math.Acos(Math.Cos(Lat1) *  
Math.Cos(Lon1) * Math.Cos(Lat2) * Math.Cos(Lon2) _
```

```
' + Math.Cos(Lat1) * Math.Sin(Lon1) * Math.Cos(Lat2) _
```

```
' * Math.Sin(Lon2) + Math.Sin(Lat1) * Math.Sin(Lat2)) *  
Radius
```

```
End Function
```

```
Public Function CalAzimuth(ByVal Lat1 As Double, ByVal  
Lon1 As Double, ByVal Lat2 As Double, ByVal Lon2 As Double)  
As Double
```

```
Dim BearingRad As Double
```

```
BearingRad = Math.Atan2((Math.Sin(Lon2) - Lon1) *  
Math.Cos(Lat2)), _
```

```
Math.Cos(Lat1) * Math.Sin(Lat2) - Math.Sin(Lat1) *  
_
```

```
Math.Cos(Lat2) * Math.Cos(Lon2 - Lon1))
```

'Radians = degrees \* (PI / 180)

'degrees = 180 \* radians/pi

CalAzimuth = 180.0 \* BearingRad / Math.PI

'Normalize) 'CalBearing = (CalBearing + 360)%360

CalAzimuth = (CalAzimuth + 360.0) Mod 360.0

End Function

Public Function CalElevation(ByVal Elev1 As Double, ByVal Elev2 As Double, ByVal Distance\_Miles As Double) As Double

CalElevation = (180 / Math.PI) \* ((Elev2 - Elev1) / Distance\_Miles - Distance\_Miles / (2 \* Radius\_Miles))

End Function

Public Function ElapsedTime(ByVal DNum As Double) As String

Dim hh, mm, ss As Integer

Dim \_hh, \_mm, \_ss As Double

\_hh = DNum \* 24

hh = Math.Truncate(\_hh)

\_mm = DecimalPart(\_hh) \* 60

mm = Math.Truncate(\_mm)

\_ss = DecimalPart(\_mm) \* 60

ss = Math.Truncate(\_ss)

ElapsedTime = hh & ":" & mm & ":" & ss

Return ElapsedTime

## Lampiran - F4

End Function

Public Function DecimalPart(ByVal nbr As Decimal) As Double

Dim wholePart, fractionalPart As Decimal

wholePart = Math.Truncate(nbr)

fractionalPart = nbr - wholePart

DecimalPart = fractionalPart

Return DecimalPart

End Function

Public Sub SaveDataGridView(ByVal dgv As DataGridView)

Dim strLine As String = ""

Using oSaveFileDialog As New SaveFileDialog

oSaveFileDialog.Title = "Save As"

oSaveFileDialog.AddExtension = True

oSaveFileDialog.DefaultExt = ".txt"

If oSaveFileDialog.ShowDialog =  
Windows.Forms.DialogResult.OK Then

Using oFile As New  
IO.StreamWriter(oSaveFileDialog.FileName, False)

strLine = "Home_Lat	Home_Lon
Home_Alt	Des_Lat
Des_Alt	Des_Lon
Elevation"	Azimuth
	Distance

oFile.WriteLine(strLine)

For index1 As Integer = 0 To dgv.RowCount - 1

```
strLine = ""  
For index2 As Integer = 0 To  
dgv.Rows(index1).Cells.Count - 1  
    strLine = strLine &  
dgv.Rows(index1).Cells(index2).Value & vbTab  
Next  
oFile.WriteLine(strLine)  
Next  
  
End Using  
End If  
End Using  
End Sub  
End Module
```



